

РОСЖЕЛДОР
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Ростовский государственный университет путей сообщения»
(ФГБОУ ВО РГУПС)

О.Г. Ведерникова

ПРОГРАММИРОВАНИЕ

Учебно-методическое пособие

Часть 2

**Средства построения многомодульных программ с
использованием функций**

Ростов-на-Дону
2016

УДК 681.3(07) + 06

Ведерникова, О.Г.

Программирование: [Электронный ресурс] учебно-методическое пособие. В 4 ч. Ч. 2. Средства построения многомодульных программ с использованием функций / О.Г. Ведерникова; ФГБОУ ВО РГУПС. – Ростов н/Д, 2016. – 88 с. –

Рассмотрены базовые приемы программирования и организации функций. Содержится лекционный материал, а также большое количество исходных текстов программ-примеров.

Приведены задания и методика выполнения лабораторных работ по дисциплине «Программирование», а также примеры выполнения лабораторных работ.

Предназначено для студентов направления подготовки 09.03.02 (230400) Информационные системы и технологии и 09.03.01 (230100) Информатика и вычислительная техника, а так же для студентов всех специальностей, изучающих дисциплину «Программирование».

Одобрено к внесению в «Электронный университет» кафедрой «Вычислительная техника и автоматизированные системы управления».

© Ведерникова О.Г., 2016
© «Электронный университет»
ФГБОУ ВО РГУПС, 2016

Оглавление

МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ	4
СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ПОДПРОГРАММ-ФУНКЦИЙ	7
ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ И ОБЛАСТИ ИХ ВИДИМОСТИ	10
ФОРМАЛЬНЫЕ И ФАКТИЧЕСКИЕ ПАРАМЕТРЫ	10
Задания для лабораторной работы №1. Организация функций пользователя	12
ФУНКЦИИ И МАССИВЫ	21
ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ СО СТРОКАМИ.....	24
ФУНКЦИИ СО СТРУКТУРАМИ	25
Задания для лабораторной работы №2. Передача массивов в функции	27
ФУНКЦИИ НЕ ВОЗВРАЩАЮЩИЕ ЗНАЧЕНИЯ	41
ПЕРЕДАЧА ПАРАМЕТРОВ ФУНКЦИИ	41
ПЕРЕДАЧА ПАРАМЕТРОВ ПО ЗНАЧЕНИЮ	41
ПЕРЕДАЧА ПАРАМЕТРОВ ПО УКАЗАТЕЛЮ.....	42
Задания для лабораторной работы №3. Особенности создания функций, не возвращающих значения	43
ПЕРЕДАЧА ПАРАМЕТРОВ ПО ССЫЛКЕ	55
ССЫЛКИ.....	55
ФУНКЦИИ СО СТРУКТУРАМИ.	58
Задания для лабораторной работы №4. Передача параметров по ссылке и по указателю	59
ПОНЯТИЕ РЕКУРСИИ.....	71
РЕКУРСИВНЫЕ ФУНКЦИИ	73
Задания для лабораторной работы №5. Рекурсивные функции.....	76
Библиографический список.....	86

МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ

Разработка больших программ включает в себя этапы: **Анализа, Планирования и Проектирование** проекта

На этапе Проектирования в первую очередь происходит структурирование задачи, то есть Разбиение общей задачи на отдельные модули (подпрограммы). в соответствии с *принципами* структурного программирования ,

Приступая к разработке сложной профессиональной программы , необходимо иметь в виду, что она, является большой системой, поэтому необходимо принять меры для ее упрощения. Для этого для упрощения программу разрабатывают по частям, которые называются подпрограммами. А сам такой метод разработки программ называют *модульным* программированием или *структурным* .

Подпрограмма – это фрагмент алгоритма, оформляемый как самостоятельный программный продукт, пригодный для многократного использования в программах. Каждая разработанная подпрограмма может включаться в состав разных программ, если выполнены условия её использования, декларированные в спецификации по этой подпрограмме. Таким образом, программный модуль может рассматриваться и как средство борьбы

со сложностью программ,

и как средство борьбы с дублированием в программировании (т.е. как средство накопления и многократного использования программистских знаний).

Программы разбиваются на модули для того, чтобы:

- упростить их разработку и реализацию;
- упростить их настройку и модификацию;
- облегчить чтение программ; облегчить работу с данными, имеющими сложную структуру;
- избежать чрезмерной детализации алгоритмов;

Использование подпрограмм или **структурирование** программы позволяет решить следующие задачи:

- 1) упростить их разработку и реализацию;
- 2) повышение надежности и облегчение тестирования и отладки
- 3) уменьшение размеров кода за счет многократно вызываемых функций из разных точек программы.
- 4) повторное (многократное) использование ранее написанного кода (*представим, что каждая новая программа писалась бы "с нуля"?!).*
- 5) решать простые подзадачи (подпрограммы) независимо несколькими группами программистов

б) экономия памяти за счет высвобождения её после выхода из подпрограммы

Не всякий программный модуль способствует упрощению программы. Выделить хороший с этой точки зрения модуль является серьезной творческой задачей.

Во время разбиения программы на подпрограммы строится дерево функций: сначала рассматривается общая функция всей программы целиком, определяются входные и выходные данные программы (форма запроса начальных данных и формы выдачи результатов); а также структура данных. (массивы, переменные, и т.д.) типы, диапазонов величин, структуры и т.п.).

Далее алгоритм разделяется на несколько подпрограмм , решающих сначала самые глобальные задачи (например такие как инициализация (ввод) данных, главная часть , завершение (вывод) данных).

Потом каждая из этих подпрограмм детализируется на более низком уровне, разбиваясь в свою очередь на небольшое число других вспомогательных подпрограмм, и так происходит до тех пор, пока вся задача не окажется реализованной. В результате получаем *дерево функций* (подпрограмм) см. рис.1.1.

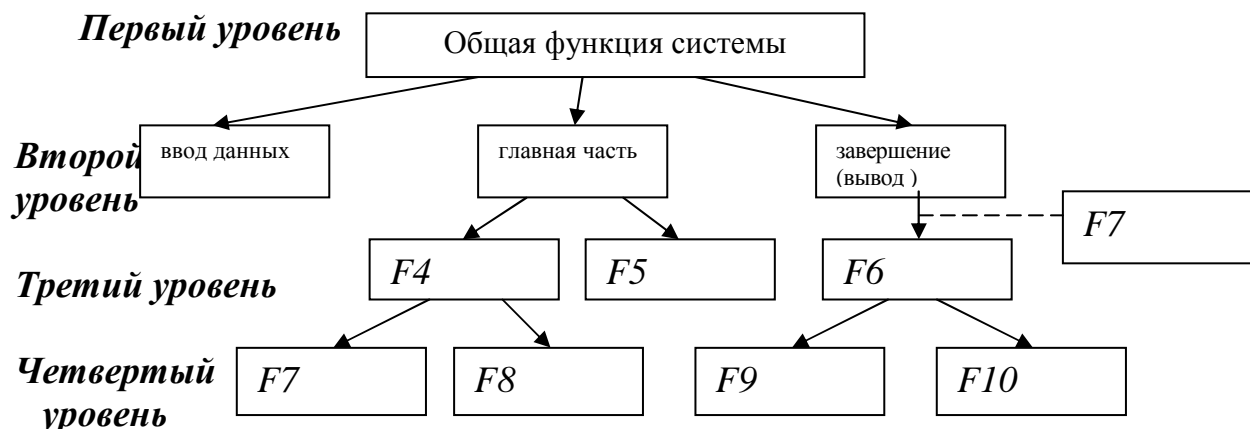


Рис. 1.1. Модульная организация программы Дерево функций (подпрограмм, модулей)

После структурирования (разбиения на подпрограммы) выполняется **спецификация** будущих подпрограмм, которая включает в себя следующие элементы:

1. Присвоение имени подпрограмме;
2. Определение входных данных и их типов
3. Определение выходных данных и их типов
4. Описание алгоритма подпрограммы;

Рассмотрим далее этап РЕАЛИЗАЦИИ (или программирования)

При нисходящем подходе «Сверху вниз» для написания и отладки текста корневого модуля (высшего уровня) тексты подчиненных модулей заменяются на заглушки.

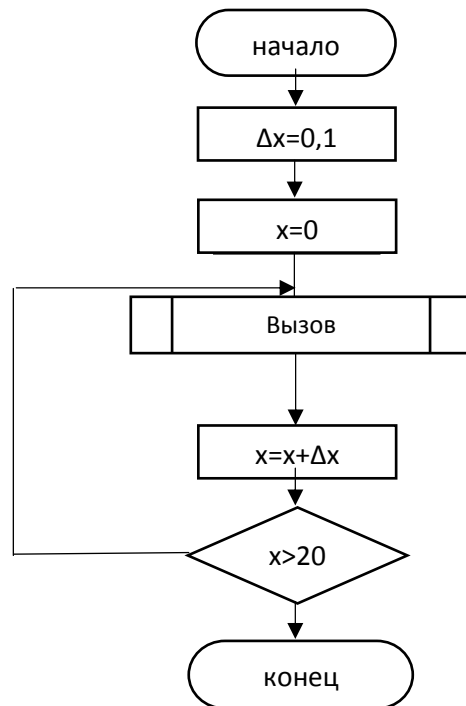
Заглушка – это примитивная модель подпрограммы. Она имеет тоже имя и тот же список параметров (спецификацию), но не реализует алгоритм (т.е. внутри только пустые begin end).

Пример заглушки, если имеется необходимость вычисления сложной математической функции, то **заглушка** реализуется одним оператором, который просто присваивает функции любое допустимое значение (например, $y:=5$)

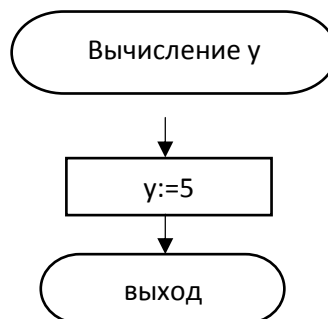
Пример создания заглушки :

Вычислить $y = \begin{cases} 2x^2 + 4, & \text{если } x < 5 \\ x - 2, & \text{если } x \geq 5 \end{cases}$ при $0 < x < 20$ с шагом $\Delta x = 0,1$

Модуль высшего уровня

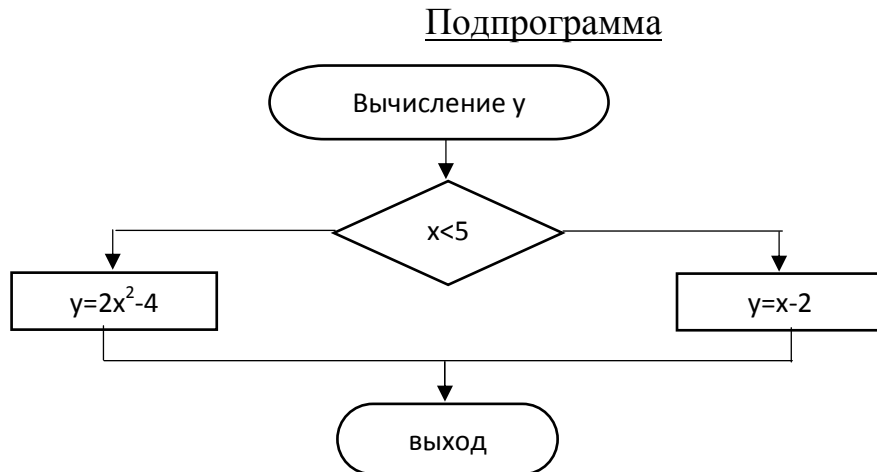


Заглушка



Полученный код основной (корневой) программы с вызовами заглушек запускается и отлаживается

Потом переходят к следующему шагу: снятию одной из заглушек. Снятие состоит в написании кода подпрограммы и его отладки. Так как порядок снятия заглушек неважен, то над проектом может работать параллельно большое количество программистов, каждый над своим модулем.



Процесс повторяется до тех пор пока не будут сняты все заглушки.

На каждом этапе создания есть возможность запустить (например, для заказчика) демонстрационную версию программы. Когда сняты все заглушки окончательной код программы готов

Подпрограммы могут быть 2 –х видов:

1) встроенные (стандартные) $\sin()$ $\cos()$ Их описание расположено в специализированных библиотечных модулях, ...

2) и определенные пользователем.

В языке C++ можно создавать 2 типа пользовательских подпрограмм-функции :

1. Подпрограммы-функции возвращающие значения
2. Подпрограммы-функции не возвращающие значения (типа void).

СОЗДАНИЕ ПОЛЬЗОВАТЕЛЬСКИХ ПОДПРОГРАММ-ФУНКЦИЙ

Любая программа на языке C/C++ содержит функцию с именем **main** (с неё начинается выполнение программы).

Пользовательские функции или функции созданные пользователем аналогичны стандартным функциям $\sin(x)$, $\cos(x)$ и т.д.

Вспомним основные термины, связанные с функциями:

$Y = \text{sqrt}(9)$; - оператор **вызова** функции

Sqrt – имя функции

9 – выражения в скобках – **аргумент функции**, параметр

3 – результат функции - **возвращаемое значение**, присваивается у
Объявление функций располагается перед объявлением главной
функции программы с именем **main**.

Формат объявления **подпрограммы-функции**:

Тип_возвращаемого_значения имя_функции (список_параметров_их_типов)

{
тело функции , описание локальных переменных, констант, типов и
т.д..

return (возвращаемое_ значение);
}

В заголовке функции указывается, во-первых, тип возвращаемого ею значения. Во-вторых, указывается имя функции, по которому будет происходить обращение к ней из другой функции. В-третьих, список принимаемых параметров или аргументов и указание их типов. Данные параметры называются формальными параметрами. Далее располагается тело функции, имеющее такую же структуру, что и функция **main()**. Оно может содержать описание локальных переменных, также исполняемые операторы и оператор **return ()**, в котором определяется возвращаемое функцией значение.

Пример:

#.....

```
Float sum(float a, float b) // a, b список формальных параметров
{
    float s ; //описание локальных переменных
    s=a+b;
    return( s ); //возвращаемое_ значение
};
```

Int main()

```
{ float x,y,sum_xy;
```

```
Sum_xy=sum(5,8);
```

```
Cout<<"сумма 1="<<sum_xy;
```

```
Cin>>x;
```

```
Cin>>y;
```

```
Sum_xy=sum(x,y);
```

```
Cout<<"сумма 2="<<sum_xy;
```

```
}
```


Пользовательскую функцию можно определить либо в том же файле, где находится главная функция `main()`, либо в отдельном файле.

Пример Задача 1.

Дано: 'n' Вычислить: $Q = \frac{5! - 6!}{(n-1)! + (n!)^2}$

Для решения данной задачи организуем подпрограмму-функцию вычисления факториала `fact(k) → k!`

// k – входной параметр

// fact – имя функции

// k! = 1*2*3*4*...*k

Выполним спецификацию подпрограммы-функции, которая включает в себя следующие элементы:

1. Имя подпрограммы - **fact**
2. Входные данные и типы – **k** – целое число, определяющее чей факториал будет вычисляться : **int**
3. Выходные данные и типы – $P=1*2*3*4*...*k$ – факториал этого числа **int**
4. Алгоритм подпрограмм – цикл `for` для накопления произведения

Program pr1;

int fact (int k)

{имя функции} {список формальных параметров} {тип результата}

*/*где **fact** – имя функции, **k**- формальный входной параметр целого типа, результат функции (то есть имя **fact**) также целого типа*/*

*{ **int p, I** ; //- это описание локальных переменных, т.е. необходимых только для внутреннего использования и невидимых (недоступных) для главной программы/*

P =1;

For (I =1;i<=k; i++) { P = P*i; };

Return(P);

}

Int main()

{**Float a, F** ; **int n** ; // глобальные переменные

Cout<<"введите n"; cin>>n;

A= fact(5); *//- оператор обращения к созданной нами функции; где в круглых скобках указано значение входного параметра, или фактический параметр, его значение подставляется вместо формального параметра }*

F = (a – fact(6))/(fact (n-1)+pow(fact(n),3); *//-это тоже вызовы функции созданной нами }*

Cout<<"f = "<<,f);

}

При вызове подпрограммы управление передается из основной программы в подпрограмму-функцию, формальным параметрам присваивается значение фактических параметров. Далее начинается выполнение операторов подпрограммы-функции. При достижении конца подпрограммы-функции управление передается обратно в точку вызова.

При вызове функции происходит:

- 1) Управление передается из главной программы в подпрограмму
- 2) При этом формальные параметры записанные в заголовке функции присваиваются фактические значения указанные в вызове функции
- 3) Начинается выполнение всех операторов записанные в подпрограмме функции. По достижению конца функции управление передается обратно в точку вызова.

ЛОКАЛЬНЫЕ И ГЛОБАЛЬНЫЕ ПЕРЕМЕННЫЕ И ОБЛАСТИ ИХ ВИДИМОСТИ

Локальными переменные – это переменные, объявленные в теле подпрограммы. Их значения доступны только во время работы подпрограммы (*и нужны только для временных внутренних работ*). При выходе из нее значения локальных переменных утрачивается.

Глобальные переменные – это те переменные, что объявлены в теле главной программы **main()**. Их значения доступны во всех подпрограммах, кроме тех, в которых объявлена локальная с таким же именем. Но не рекомендуется изменять значения глобальных переменных в подпрограммах поскольку затрудняет отладку программы.

ФОРМАЛЬНЫЕ И ФАКТИЧЕСКИЕ ПАРАМЕТРЫ

Подпрограмма взаимодействует с основной программой через **механизм параметров** – это входные и выходные данные, с которыми работает подпрограмма.

Формальные параметры – это параметры, определенные в заголовке *описания* подпрограммы.

Фактические параметры – это выражения, задающие конкретные значения *при вызове* подпрограммы из главной. То есть при обращении к подпрограмме ее формальные параметры заменяются фактическими, переданными из главной программы.

Название «формальные» эти параметры получили в связи с тем, что они задают только имена для обозначения исходных данных и результатов (это пустые ячейки). При вызове же подпрограммы на их место будут подставлены конкретные значения.

Само по себе описание (*или декларация*) процедуры или функции еще не вызывает выполнения никаких действий.

Чтобы подпрограмма *процедура* сработала, ее нужно *вызвать* из главной программы, {A:=fact (5)}. записав в нужной точке программы имя подпрограммы- процедуры со *списком фактических параметров*, которые будут подставлены *на место формальных*.

При этом работа главной программы на некоторое время приостанавливается, и начинает выполняться вызванная подпрограмма.

В этот момент В момент вызова функции в оперативной памяти выделяется отдельная временный область памяти для хранения локальных переменных этой подпрограммы на время ее работы. Туда же временно помещаются копии фактических параметров.

По завершении подпрограммы функция возвращает результат главной программе. А также освобождается временная область памяти, где были записаны локальные переменные и их значения утрачиваются.

Заметим, что имена формальных и фактических параметров могут совпадать. Это не приводит к проблемам, так как соответствующие им переменные все равно будут различны из-за того, что хранятся в разных областях памяти. Кроме того, все формальные параметры(и локальные переменные) являются временными переменными - они создаются в момент вызова подпрограммы и уничтожаются в момент выхода из нее.

Таким образом при модульном программировании происходит экономия оперативной памяти. то есть нет необходимости выделять память одновременно для всех переменных, участвующих в вычислениях.

Каждый новый вызов модуля порождает новую копию набора локальных данных и размещение их в отдельно выделенной области памяти

При разделении на подпрограммы необходимо, учитывая следующие требования *структурного программирования*:

Основные принципы структурного программирования

1. Подпрограмма должна иметь минимальную возможность обмена информацией с *другими подпрограммами* и с главной программой.

2. Подпрограмма может использовать глобальные переменные , что не рекомендуется, но не должна их никогда изменять (*без крайней необходимости*).

3. Подпрограмма может обмениваться данными с «внешним миром» (то есть с главной программой) только через свои параметры.

4. Подпрограмма не может запрашивать у пользователя входные данные или выводить результаты на экран пользователя. Только если это не специально организованная подпрограмма для ввода или вывода.

5. Подпрограмма должна быть организована так, что бы её можно было вызывать многократно с различными наборами значений входных параметров из разных точек основной программы, и тогда она будет

способна решать целый класс схожих задач. Такие подпрограммы дают Большой эффект

Задания для лабораторной работы №1. Организация функций пользователя

Задание №1

1. Создать функцию, которая возвращает меньшее из двух данных чисел. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

2. Создать функцию, которая переводит время, заданное в минутах в секунды. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

3. Создать функцию, которая определяет периметр треугольника по трем его сторонам. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

4. Создать функцию, которая возвращает номер квадранта, в котором находится точка. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

5. Создать функцию, которая возвращает среднее арифметическое трех данных чисел. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

6. Создать функцию, которая определяет площадь круга по его

радиусу. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

7. Создать функцию, которая возвращает остаток от деления двух натуральных чисел. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

8. Создать функцию, которая переводит радианы в градусы. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

9. Создать функцию, которая определяет длину отрезка по его координатам. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

10. Создать функцию, которая возвращает в долларах сумму, заданную в рублях. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

11. Создать функцию, которая возвращает большее из двух данных чисел. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

12. Создать функцию, которая определяет длину окружности по заданному радиусу. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать

созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

13. Создать функцию, которая переводит скорость из км/час в м/сек. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

14. Создать функцию, которая возвращает среднее геометрическое двух данных чисел. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

15. Создать функцию, которая возвращает в рублях сумму, заданную в долларах. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

16. Создать функцию, которая переводит градусы в радианы. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

17. Создать функцию, которая переводит время, заданное в секундах в минуты. Для создаваемой функции: подобрать имя; указать тип функции; выбрать имена и типы входных параметров; описать тело функции с *обязательным* оператором в конце; в главной программе вызвать созданную функцию два раза с различными входными данными. Вывести результаты в главной программе.

Задание №2 Функция, возвращающая результат

Вариант № 1 Реализовать функцию. Функция вычисляет площадь параллелограмма $s = a b \cos \alpha$ по заданным двум сторонам и углу между ними. В главной программе задано два параллелограмма. Найти их площади, вызвав функцию 2 раза.

Вариант №2 Реализовать функцию . Функция вычисляет площадь квадрата по заданной стороне. В главной программе задано два квадрата. Найти их площади , вызвав функцию 2 раза.

Вариант №3 Реализовать функцию . Функция вычисляет диагональ прямоугольника по заданным двум сторонам . В главной программе задано два прямоугольника . Найти их диагонали, вызвав функцию 2 раза.

Вариант №4 Реализовать функцию . Функция вычисляет площадь $s = \frac{1}{2}(a + b) h$. равнобедренной трапеции по заданным двум основаниям a, b и высоте h . ..В главной программе задано две равнобедренные трапеции . Найти их площади, вызвав функцию 2 раза.

Вариант № 5. Реализовать функцию. Функция вычисляет площадь окружности по заданному радиусу R . В главной программе задано две окружности . Найти их площади , вызвав функцию 2 раза.

Вариант № 6. Реализовать функцию . Функция вычисляет объем сферы $v = \frac{4}{3} \pi R^3$ по заданному радиусу R . В главной программе задано две сферы. Найти их объемы, вызвав функцию 2 раза.

Вариант № 7. Реализовать функцию . Функция вычисляет объем квадратной призмы по заданной высоте H и стороне основания a . В главной программе задано две квадратные призмы . Найти их объемы , вызвав функцию 2 раза.

Вариант № 8. Реализовать функцию . Функция вычисляет объем правильной треугольной призмы по заданной высоте H и стороне основания a . .В главной программе задано две треугольные призмы . Найти их объемы , вызвав функцию 2 раза.

Вариант №9 Реализовать функцию . Функция вычисляет объем цилиндра по заданной высоте H и радиусу основания R . В главной программе задано два цилиндра . Найти их объемы , вызвав функцию 2 раза.

Вариант №11 Реализовать функцию . Функция вычисляет площадь боковой поверхности $S = \pi R \sqrt{R^2 + H^2}$ конуса по заданной высоте H и радиусу основания R . В главной программе задано два конуса . Найти их площади боковых поверхностей, вызвав функцию 2 раза.

Вариант №11 Реализовать функцию . Функция вычисляет объем $v = \frac{1}{3} H \pi R^2$ конуса по заданной высоте H и радиусу основания R . В главной программе задано два конуса . Найти их объемы, вызвав функцию 2 раза.

Вариант №13. Реализовать функцию . Функция вычисляет объем $v=1/3$ $H \pi (R^2 + r^2 + R r)$ Усеченного конуса по заданной высоте H и двум радиусам оснований R и r . .В главной программе задано два конуса . Найти их объемы , вызвав функцию 2 раза.

Вариант №14 Реализовать функцию . Функция вычисляет длину вектора на плоскости по заданным координатам x и y . В главной программе задано два вектора . Найти их длины, вызвав функцию 2 раза.

Вариант №15 Реализовать функцию . Функция вычисляет тангенс угла наклона (между осью Ox и вектором) вектора на плоскости по заданным координатам x и y . В главной программе задано два вектора . Найти их углы, вызвав функцию 2 раза.

Вариант №16. Реализовать функцию . Функция вычисляет расстояние до начала координат от точки на плоскости по заданным её координатам x и y . В главной программе задано две точки . Найти их расстояния до начала координат , вызвав функцию 2 раза.

Задача 3

1. **задача 2.** Описать функцию $\text{Sign}(X)$ целого типа, возвращающую для вещественного числа X его знак , то есть следующие значения: -1 , если $X < 0$, 0 , если $X = 0$; 1 , если $X > 0$. С помощью этой функции найти значение выражения $\text{Sign}(A) + \text{Sign}(B)$ для данных вещественных чисел A и B .

2. **задача 2** Описать функцию $\text{RootsCount}(A, B, C)$ целого типа, определяющую количество корней квадратного уравнения $A*x^2 + B*x + C = 0$ (A, B, C — вещественные параметры, $A \neq 0$). С ее помощью найти количество корней для каждого из трех квадратных уравнений с данными коэффициентами. Количество корней определять по значению дискриминанта: $D = B^2 - 4*A*C$.

3. **задача 2** Описать функцию $\text{CircleS}(R)$ вещественного типа, находящую площадь круга радиуса R (R — вещественное). С помощью этой функции найти площади трех кругов с данными радиусами. Площадь круга радиуса R вычисляется по формуле $S = p*R^2$. В качестве значения p использовать 3,14. ,

4. **задача 2** Описать функцию $\text{RingS}(R1, R2)$ вещественного типа, находящую площадь кольца, заключенного между двумя окружностями с общим центром и радиусами $R1$ и $R2$ ($R1$ и $R2$ — вещественные, $R1 > R2$). С ее помощью найти площади трех колец, для которых даны внешние и внутренние радиусы. Воспользоваться формулой площади круга радиуса R : $S = p*R^2$. В качестве значения p использовать 3.14.

5. **задача 2** Описать функцию $\text{TriangleP}(a, h)$, находящую периметр равнобедренного треугольника по его основанию a и высоте h , проведенной к основанию (a и h — вещественные). С помощью этой, функции найти периметры трех треугольников, для которых даны основания и высоты. Для нахождения боковой стороны b треугольника использовать теорему

Пифагора: $b^2 = (a/2)^2 + h^2$.

6. **задача 2** Описать функцию Calc(A,B,Op) вещественного типа, выполняющую над ненулевыми вещественными числами A и B одну из арифметических операций и возвращающую ее результат. Вид операции определяется целым параметром Op : 1 — вычитание, 2 — умножение, 3 — деление, остальные значения — сложение. С помощью Calc выполнить для данных A и B операции, определяемые данными целыми $N1, N2, N3$.

7. **задача 2** Описать функцию Quarter (x, y) целого типа, определяющую номер координатной четверти, в которой находится точка с ненулевыми вещественными координатами (x, y). С помощью этой функции найти номера координатных четвертей для трех точек с данными ненулевыми координатами.

8. **задача 2.** Вычисление скалярного произведения векторов (массивов) оформить в виде функции (x, y)- скалярное произведение векторов — это сумма поэлементных произведений их компонент. В главной программе заданы два вектора (массива) $x = (x_1, x_2, x_3, x_4)$, и $y = (y_1, y_2, y_3, y_4)$. Определить косинус угла α между векторами x и y по формуле: $\cos \alpha = \frac{(x, y)}{\sqrt{(x, x)(y, y)}}$, где $(x, y) = x_1y_1 + x_2y_2 + \dots + x_ny_n$ — скалярное произведение векторов (сумма поэлементных произведений компонент), используя описанную функцию 3 раза.

9. **задача 2.** Даны длины сторон треугольника a, b, c . Найти медианы треугольника, сторонами которого являются медианы исходного треугольника. Для вычисления медианы проведенной к стороне a , использовать формулу $0,5 \cdot \sqrt{2b^2 + 2c^2 - a^2}$ Вычисление медианы оформить в виде функции.

10. **задача 2.** Описать функцию DegToRad(D) вещественного типа, находящую величину угла в радианах, если дана его величина D в градусах (D — вещественное число, $0 < D < 360$). Воспользоваться следующим соотношением: $180^\circ = \pi$ радианов. В качестве значения π использовать 3.14. С помощью функции DegToRad перевести из градусов в радианы пять данных углов.

11. **задача 2.** Описать функцию RadToDeg(R) вещественного типа, находящую величину угла в градусах, если дана его величина R в радианах (R — вещественное число, $0 < R < 2\pi$). Воспользоваться следующим соотношением: $180^\circ = \pi$ радианов. В качестве значения π использовать 3.14. С помощью функции RadToDeg перевести из радианов в градусы пять данных углов.

12. **задача 2.** Четыре точки заданы своими координатами $X(x1, x2)$, $Y(y1, y2)$, $Z(z1, z2)$, $P(p1, p2)$. Выяснить, какие из них находятся на максимальном расстоянии друг от друга и вывести на печать значение этого расстояния. Вычисление расстояния между двумя точками оформить в виде функции.

13. **задача 2.** Четыре точки заданы своими координатами $X(x_1, x_2, x_3)$, $Y(y_1, y_2, y_3)$, $Z(z_1, z_2, z_3)$, $T(t_1, t_2, t_3)$. Выяснить, какие из них находятся на минимальном расстоянии друг от друга и вывести на печать значение этого расстояния. Вычисление расстояния между двумя точками оформить в виде функции.

14. Описать функцию $\text{Power2}(A, N)$ вещественного типа, находящую величину A^N (A — вещественный, N — целый параметр) по следующим формулам:

$A^0=1$ $A^N = A * A * \dots * A$ (N сомножителей), если $N > 0$; $A^N = 1 / (A * A * \dots * A)$ ($|N|$ сомножителей), если $N < 0$. С помощью этой функции найти A^K , A^L , A^M , если даны числа A , K , L , M .

Задача № 4 "функции"

Для всех вариантов: в функциях не должно быть операторов ввода или вывода

1. **Задача 3.** Даны действительные числа a , b , c . Получить:

$$\frac{\max(a, a+b) + \max(a, b+c)}{1 + \max(a+b \cdot c, b \cdot 15)}$$

Описать функции нахождения наибольшего числа из 2 заданных величин.

2. **задача 3.** Описать функцию $\text{Power1}(A, B)$ вещественного типа, находящую величину A^B по формуле $A^B = \exp(B * \ln(A))$ (параметры A и B — вещественные). В случае нулевого или отрицательного параметра A функция должна возвращать 0. С помощью этой функции в главной программе найти степени A^P , B^P , C^P , если даны числа P , A , B , C .

3. **Задача 3.** Даны действительные числа a , b . Получить $u = \min(a, b-a)$, $y = \min(ab, a+b)$, $k = \min(u+y^2, 3.14)$. Описать функции нахождения минимального числа из 2 заданных величин.

4. **Задача 3.** Даны действительные числа s , t . Получить:
 $g(1.2, s) + g(t, s) - g(2s-1, 5t)$, где $g(a, b) = \frac{a^2 + b^2}{a^2 + 2ab + 3b^2 + 4}$ - функция

5. **Задача 3.** Даны действительные числа x , y . Получить:
 $f(x, -2y, 1.17) + f(2.2, x, x-y)$, где $f(a, b, c) = \frac{2a - b - \sin c}{5 + |c|}$ - функция

6. **Задача 3.** Даны натуральные числа a , b , c . Найти НОД(a , b , c), используя формулу: $\text{НОД}(a, b, c) = \text{НОД}(\text{НОД}(a, b), c)$. Алгоритм Евклида: $\text{НОД}(A, B) = \text{НОД}(B, A \bmod B)$, если $B \neq 0$; $\text{НОД}(A, 0) = A$. Описать функцию $\text{НОД}(A, B)$, используя цикл while.

7. **Задача 3.** Получить для $x = 1, 3, 4, 5$ все значения выражения $Q(x) = p(x+1) - p(x)$ вывести на экран, где $p(y) = a_3 y^3 + a_2 y^2 + a_1 y + a_0$; - функция (где a_3, a_2, a_1, a_0 - определенные константы)

8. **Задача 3.** Даны действительные числа x , y . Получить:

$Q = \text{tg}(f(x+y, xy, y-x) + f(3.1, 1.4, y - \sin x))$, где $f(a, b, c) = \frac{3b + c - e^{-a}}{1 + |\cos 3a|}$ - функция

9. **Задача 3.** Даны действительные числа a, b . Получить $u = \min(a, b * a)$, $y = \min(a - b, a + b)$, $k = \min(u^3 + y^3, 28)$. Описать функции нахождения минимального числа из 2 заданных величин.

10. **Задача 3.** Даны действительные числа a, b . Получить $r = \max(a, b + a)$, $d = \max(ab, a + b)$, $s = \max(r + d^2, 3.14)$. Описать функции нахождения наибольшего числа из 2 заданных величин.

11. **Задача 3.** Даны действительные числа a_0, a_1, a_2, a_3 . Получить для $x = 2, 4, 7$ значения $Q(x) = p(x+1) + p(x)$, где $p(y) = a_3 y^2 + a_2 y + 2(a_1 + a_0)$. - функция

12. **Задача 3.** Даны действительные числа s, t . Получить:
 $Q = |g(\ln(s, t+1)) - g(t, s)|$, где $g(a, b) = \frac{a^2 + b^2}{a^2 + 2ab + 3b^2 + 4}$ - функция.

13. **Задача 3.** Даны натуральные числа a, b, c . Найти НОД(a, b, c), используя формулу: $\text{НОД}(a, b, c) = \text{НОД}(\text{НОД}(a, b), c)$.
Алгоритм Евклида: $\text{НОД}(A, B) = \text{НОД}(B, A \bmod B)$, если $B \neq 0$;
 $\text{НОД}(A, 0) = A$ Описать функцию НОД(A, B), используя цикл while.

14. **задача 3.** Три точки заданы своими координатами $X(x_1, x_2)$, $Y(y_1, y_2)$ и $Z(z_1, z_2)$. Найти и напечатать координаты точки, для которой угол между осью абсцисс и лучом, соединяющим начало координат с точкой, минимальный. Вычисление угла оформить в виде функции по формуле $\alpha = \arctg\left(\frac{y}{x}\right)$. Вызвать функцию в программе 3 раза.

Дополнительная * Задача 1*. Логические функции

1. **задача 5** *Описать функцию IsPower5(K) логического типа, возвращающую True, если целый параметр $K (> 0)$ является степенью числа 5, и False в противном случае. С ее помощью найти количество степеней числа 5 в наборе из 10 целых положительных чисел.

2. **задача 5** *Описать функцию IsPowerN(K, N) логического типа, возвращающую True, если целый параметр $K (> 0)$ является степенью числа $N (> 1)$, и False в противном случае. Дано число $N (> 1)$ и набор из 10 целых положительных чисел. С помощью функции IsPowerN найти количество степеней числа N в данном наборе.

3. **задача 5** *Описать функцию IsPrime(N) логического типа, возвращающую True, если целый параметр $N (> 1)$ является простым числом, и False в противном случае (число, большее 1, называется *простым*, если оно не имеет положительных делителей, кроме 1 и самого себя). Дан набор из 10 целых чисел, больших 1. С помощью функции IsPrime найти количество простых чисел в данном наборе.

4. **задача 5.** *Даны отрезки a, b, c и d . Для каждой тройки этих отрезков, из которой можно построить треугольник, вывести на экран

площадь данного треугольника. Проверку существования треугольника оформить в виде функции логического типа (или возвращающей 1 в случае «да», и 0 в случае «нет».)

5. **задача 5.*** Описать функцию $IsPalidrom(K)$, возвращающую True, если целый параметр $K (> 0)$ является *палиндромом* (то есть его запись читается одинаково слева направо и справа налево), и False в противном случае. С ее помощью найти количество палиндромов в наборе из 10 целых положительных чисел. При описании функции можно использовать функции DigitCount и DigitN из заданий Pгoc29 и Pгoc30.

6. **задача 5 *** Описать функцию $Even(k)$ логического типа, возвращающую True, если целый параметр K является четным, и False в противном случае. С ее помощью найти количество четных чисел в наборе из 10 целых чисел.

7. **задача 5 *** Описать функцию $IsSquare(K)$ логического типа, возвращающую True, если целый параметр $K (> 0)$ является квадратом некоторого целого числа, и False в противном случае. С ее помощью найти количество квадратов в наборе из 10 целых положительных чисел.

8. **задача 5 *** Задана окружность $(x-a)^2 + (y-b)^2 = R^2$ и точки $P(p1, p2)$, $F(f1, f2)$, $L(l1, l2)$. Выяснить и напечатать, сколько точек лежит внутри окружности. Проверку, лежит ли точка внутри окружности, оформить в виде функции логического типа (или возвращающей 1 в случае «да», и 0 в случае «нет».)

9. **задача 5 *** Описать функцию $IsLeapYear(Y)$ логического типа, которая возвращает True, если год Y (целое положительное число) является високосным, и False в противном случае. Вывести значение функции $IsLeapYear$ для пяти данных значений параметра Y . *Високосным* считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400.

10. **задача 5 *** Описать логическую функцию, проверяющую, можно ли путем перестановок символов в строке $S1$ получить строку $S2$. Считать, что обе строки одной длины.

11. **задача 5 *** Описать функцию $IsPowerN(K,N)$ логического типа, возвращающую True, если целый параметр $K (> 0)$ является степенью числа $N (> 1)$, и False в противном случае. Дано число $N (> 1)$ и набор из 10 целых положительных чисел. С помощью функции $IsPowerN$ найти количество степеней числа N в данном наборе.

12. **задача 5.** *Описать функцию $IsPalidrom(K)$, возвращающую True, если целый параметр $K (> 0)$ является *палиндромом* (то есть его запись читается одинаково слева направо и справа налево), и False в противном случае. С ее помощью найти количество палиндромов в наборе из 10 целых положительных чисел. При описании функции можно использовать функции DigitCount и DigitN из заданий Pгoc29 и Pгoc30.

13. **задача 5 *** Описать функцию $IsPrime(N)$ логического типа, возвращающую True, если целый параметр $N (> 1)$ является простым числом,

и False в противном случае (число, большее 1, называется *простым*, если оно не имеет положительных делителей, кроме 1 и самого себя). Дан набор из 10 целых чисел, больших 1. С помощью функции IsPrime найти количество простых чисел в данном наборе.

14. **задача 5** *Описать функцию IsSquare(K) логического типа, возвращающую True, если целый параметр $K (> 0)$ является квадратом некоторого целого числа, и False в противном случае. С ее помощью найти количество квадратов в наборе из 10 целых положительных чисел.

15. **задача 5** *Описать функцию IsLeapYear(Y) логического типа, которая возвращает True, если год Y (целое положительное число) является високосным, и False в противном случае. Вывести значение функции IsLeapYear для пяти данных значений параметра Y . Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400.

ФУНКЦИИ И МАССИВЫ

При использовании массива в качестве параметра функции в подпрограмму передается не весь массив, а только указатель на его нулевой элемент (*вы уже знаете , что имя массива это указатель на его нулевой элемент*). При этом информация о количестве элементов массива теряется, и следует передавать его размерность через отдельный параметр (или использовать глобальные константы)

В этом случае значения массива могут быть изменены внутри функции и эти изменения доступны в главной программе.

В операторе вызова функции в скобках указывается имя массива без индекса, то есть указатель на нулевой элемент

Пример **Задача2.**

Даны 3 массива : a,b,c. Вычислить:

$$Q = \frac{\min(a) + \min(b)}{\min(a - c) - \min(c)}$$

или

$Q = (\min(a_i) + \min(b_i)) / (\min(a_i - c_i) - \min(c_i))$). Описать функцию, облегчающую вычисление Q , то есть функцию нахождения минимального элемента массива.

Выполним **спецификацию** подпрограммы-функции,

1. Имя подпрограммы - **minimum**
2. Входные данные и типы – **x**: – массив из n вещественных элементов ,
3. Выходные данные и типы – **min** - минимальный элемент – вещественное число

4. Алгоритм подпрограмм – цикл for для нахождения мин. в массиве

```
float minimum( float *x, int n)
/* или ( float x[], int n)или( float x[10], int n) тогда - 10 игнорируется*/
{ float min=1000;//min=x[0];
  for (int i=0;i<n;i++) if (x[i]<min)min=x[i];
  return(min);
}
```

*void vvod (float x[], int n){ for(int i=0;i<n;i++){x[i]=rand()%5-2;}}/*так как параметр передается по указателю, то изменения в массиве доступны в главной программе*/*

```
void vyvod (float x[], int n, string name)
{ for( int i=0;i<n;i++){cout<<name<<" ["<<i<<"]="<<x[i]<<endl; }}
```

int main()//начало главной программы
 { float a[5],b[7],c[5],f[5]
 // - описание 3-х исходных массивов (и одного вспомогательного массива *f*)

```
vvod(a,5); vvod(b,7);
```

```
vvod(c,5);
```

```
vyvod(a,5,"a ");vyvod(b,7,"b ");
```

```
vyvod(c,5,"c ");
```

```
for (int i=0;i<5;i++){a[i]=rand()%10-5;c[i]=rand()%20-10;}
```

```
for (int i=0;i<7;i++){b[i]=rand()%10-5;}
```

float R= minimum (a,5) + minimum (b,7);// - вызов функции в скобках указываем имя массива без индекса

for (int i=0;i<5;i++) { f[i]=a[i]-c[i]; } /- так как массивы вычитаются поэлементно в цикле, то перед следующим вызовом функции для вычисления **min(a_i-c_i)** необходимо создать новый массив *f*, равный поэлементной разности *a* и *c* в цикле*/*

```
vyvod(f,5,"f ");
```

```
float Q = R/(minimum(f,5) – minimum(c,5));
```

```
cout<<Q;
```

```
}
```

ФУНКЦИИ, ВОЗВРАЩАЮЩИЕ УКАЗАТЕЛЬ

Пример

```
float * vvod( int n)
```

```
{
```

```
float *y;
```

```
for (int i=0;i<n;i++) y[i]=rand()%11-5;
```

```
return(y);
```

```

}
int main()
{.....
    float a[10],b[m]...;
    a=vvod(10);
    b=vvod(m);
    ...}

void vvod (float x[], int n)
{
for( int i=0;i<n;i++)
{x[i]=rand()%5-2;}
}

```

Пример:

Дано 3 одномерных массива a , b , d длиной 20 элементов каждый. Описать подходящую функцию для облегчения решения задачи. Вычислить

$$V = \begin{cases} \prod_{i=1}^{20} a_i - \prod_{i=1}^{20} (b_i + d_i), & \text{если } \prod_{i=1}^{20} a_i > \prod_{i=1}^{20} d_i; \\ \prod_{i=1}^{20} (b_i - a_i) + \prod_{i=1}^{20} d_i, & \text{иначе;} \end{cases}$$

```

#include<stdio.h>
#include<conio.h>
#include<math.h>
#include<iostream.h>
#include<time.h>
#include<stdlib.h>
float proizved(float *x, int n)//или (float x[], int n) или (float x[20], int n)
тогда - 20 игнорируется
{float P=1;
for (int i=0;i<n;i++) { P*=x[i]; }
return(P);
}
void vvod (float x[], int n){ for( int i=0;i<n;i++){x[i]=rand()%5-2;}}/*так
как параметр передается по указателю, то изменения в массиве доступны в
главной программе*/

```

```

void vyvod (float x[], int n, string name)
{ for( int i=0;i<n;i++){cout<<name<<" ["<<i<<"]="<<x[i]<<endl; }}

```

```

int main()
{
float a[20],b[20],d[20],c[20],r[20],V;

```

```

vvod(a,20);
vvod(b,20);
vvod(d,20);
for (int i=0;i<20;i++)
{ c[i]=b[i]+d[i]; r[i]=b[i]-a[i]; }
vyvod(a,5,"a ");
vyvod(b,5,"b ");
vyvod(c,5,"c ");
vyvod(d,5,"d ");
vyvod(r,5,"r ");

if (proizved(a,20)>proizved(d,20)) V=proizved(a,20)-proizved(c,20);
else
V=proizved(r,20)+proizved(d,20);
cout<<"V="<<V;
getch();
}

```

Если описание функции расположить в отдельном файле или в этом же файле, но после main(), то необходимо дополнительно записать строку или прототип функцию, расположенную после подключения библиотек и до описания главной функции.

ПОЛЬЗОВАТЕЛЬСКИЕ ФУНКЦИИ СО СТРОКАМИ

Пример **Вычислить количество указанных символов в строке**

```

#include<conio.h>
#include<string.h>
#include<iostream>
using namespace std;
int count( string st,char cim)
{
    int k=0;
    for(int i=0;i<st.length();i++)
    {
        if(st[i]==cim) k++;
    }
    return(k);
}

```

```

int main ()
{string sta,spb;int n1,n2;
    cout<<"vvedite tekst=";

```



```

getline(cin,sta);//ВВОД строки
cout<<"tekst="<<sta<<endl;
int kk= count(sta,'j');
cout<<endl<<"kk="<<kk;
getch();
}

```

Пример **Убрать из строки указанные символы**

```

#include<conio.h>
#include<string.h>
#include<iostream>
using namespace std;
string remove( string st, char cim)
{
    string tx="";
    for(int i=0;i<st.length();i++)
    {
        if(st[i]!=cim) tx+=st[i];
    }
    return(tx);
}
int main ()
{
    string sta,spb;int n1,n2;
    cout<<" vvedite tekst=";
    getline(cin,sta);//ВВОД строки
    cout<<" ishodnyi tekst="<<sta<<endl;//ВЫВОД строки

    string tx_a=remove(sta,'y');
    cout<<endl<<"text izmenenyi="<<tx_a;

    getch();
}

```

ФУНКЦИИ СО СТРУКТУРАМИ

Пример Функция поиска названия высочайшей горы

```

#include<stdlib.h>
#include<time.h>
#include<conio.h>
#include<math.h>
#include<iostream.h>
#include<time.h>
#include<stdlib.h>

```

```

#include<string.h>
using namespace std;
struct T_mountain
{
    string name;
    int h;
};

string max_gora (T_mountain ro[], int n)
{
    int max = 0;
    string max_name="";
    for (int i = 0; i < n; i ++ )
    {
        if (ro[i].h>max)
        {
            max= ro[i].h;
            max_name=ro[i].name;
        }
    }
    return max_name;
}

int main()
{

    struct T_mountain rocks[10];
    int n, i;
    cout<<"vvedine n";
    cin >>n;
    for (i = 0; i < n; i ++ )
    { cout<<"vvedite name";
      cin>>rocks[i].name ;
      cout<<"vvedite height";
      cin>>rocks[i].h ;
    }
    string a=max_gora(rocks, n);
    cout<< "imya vysokoi gory="<< a;
    getch();
}

```

Задания для лабораторной работы №2. Передача массивов в функции

Задача 1. «Функции с Массивами»

Массивы должны заполняться значениями в главной программе или в дополнительных подпрограммах

1. Оформить функцию поиска количества отрицательных элементов массива. В главной программе дано 3 одномерных массива a,b,c длиной 10 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

2. Оформить функцию поиска количества положительных элементов массива. В главной программе дано 2 одномерных массива x,y длиной 20 элементов каждый и один массив z длиной 5 элементов.. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

3. Оформить функцию поиска количества элементов равных 5 . В главной программе дано 3 одномерных массива p,q,r длиной 10 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

4. Оформить функцию поиска количества нулевых элементов массива. В главной программе дано 3 одномерных массива arr1,arr2,arr3 длиной 10 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

5. Оформить функцию поиска количества элементов массива, больших заданного числа α . В главной программе дано 2 одномерных массива a,b длиной 15 элементов каждый. Применить функцию для каждого из 2-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

6. Оформить функцию поиска суммы отрицательных элементов массива. В главной программе дано 2 одномерных массива x, y длиной 10 элементов каждый и один массив z длиной 5 элементов.. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

7. Оформить функцию поиска суммы элементов массива больших 5. В главной программе дано 2 одномерных массива a,b,c длиной 10 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

8. Оформить функцию поиска суммы элементов массива больших 1. В главной программе дано 2 одномерных массива mass1, mass2 длиной 10 элементов каждый и один массив z длиной 5 элементов. Применить функцию

для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

9. Оформить функцию поиска суммы элементов массива, больших заданного числа α . В главной программе дано 4 одномерных массива a, b, c, d длиной 10 элементов каждый. Применить функцию для каждого из 4-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

10. Оформить функцию поиска суммы элементов массива, больших заданного числа α и меньшего заданного числа β . В главной программе дано 3 одномерных массива a, b, c длиной 10 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

11. Оформить функцию поиска произведения положительных элементов массива. В главной программе дано 4 одномерных массива a, b, c, d длиной 10 элементов каждый. Применить функцию для каждого из 4-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

12. Оформить функцию поиска произведения отрицательных элементов массива. В главной программе дано 4 одномерных массива x, y, z, f длиной 10 элементов каждый. Применить функцию для каждого из 4-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

13. Оформить функцию поиска произведения элементов массива, больших заданного числа α и меньшего заданного числа β . В главной программе дано 2 одномерных массива a, b длиной 10 элементов каждый и один массив z длиной 5 элементов. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

14. Оформить функцию поиска среднего арифметического отрицательных элементов массива. В главной программе дано 3 одномерных массива a, b, c длиной 10 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

15. Оформить функцию поиска среднего арифметического положительных элементов массива. В главной программе дано 3 одномерных массива a, b, c длиной 10 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

16. Оформить функцию поиска **среднего** геометрического элементов массива. В главной программе дано 3 одномерных массива a,b,c длиной 10 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

17. Оформить функцию поиска **арифметического** элементов массива **меньшего** заданного числа β . В главной программе дано 3 одномерных массива a,b,c длиной 10 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. (в функции не должно быть операторов ввода или вывода)

Задача 2. «Функции с Массивами»

Массивы должны заполняться значениями в главной программе или в дополнительных подпрограммах.

18. . Оформить функцию поиска **суммы** элементов, стоящих на нечетных местах (*использовать шаг цикла $\neq 1$*), В главной программе дано 3 одномерных массива a,b,c длиной 30 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. Найти произведение найденных сумм элементов. (в функции не должно быть операторов ввода или вывода)

19. Оформить функцию поиска **максимального** элемента в одномерном массиве, В главной программе Дано 3 одномерных массива a,b,c длиной 20 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. Найти сумму найденных максимальных элементов. (в функции не должно быть операторов ввода или вывода)

20. Оформить функцию поиска **номера** последнего **нулевого** элемента в массиве, В главной программе Дано 3 одномерных массива a,b,c длиной 20 элементов каждый. Применить функцию для каждого из 3-х заданных массивов. Найти сумму найденных номеров элементов. (в функции не должно быть операторов ввода или вывода)

21. . Оформить функцию поиска **минимального** элемента среди положительных в одномерном массиве, В главной программе Дано 4 одномерных массива a,b,c,d длиной 10 элементов каждый. применить функцию для каждого из 4-х заданных массивов. найти сумму найденных минимальных элементов. (в функции не должно быть операторов ввода или вывода)

22. Оформить функцию поиска **номера** последнего **отрицательного** элемента в массиве, В главной программе Дано 3 одномерных массива a,b,c длиной 20 элементов каждый. применить функцию для каждого из 3-х заданных массивов. найти разность найденных номеров элементов. (в функции не должно быть операторов ввода или вывода)

23. . Оформить функцию поиска **минимального** элемента среди

элементов, стоящих на **четных** местах (*использовать шаг цикла $\neq 1$*)), В главной программе Дано 4 одномерных массива a,b,c,d длиной 10 элементов каждый. применить функцию для каждого из 4-х заданных массивов. найти произведение найденных минимальных элементов. (в функции не должно быть операторов ввода или вывода)

24. Оформить функцию поиска **номера** последнего **положительного** элемента в массиве, В главной программе Дано 2 одномерных массива a,b, длиной 15 элементов каждый. применить функцию для каждого из 2-х заданных массивов. найти произведение найденных номеров элементов. (в функции не должно быть операторов ввода или вывода)

25. Оформить функцию поиска **максимального** элемента массива среди элементов, стоящих на **нечетных** местах (*использовать шаг цикла $\neq 1$*)). В главной программе Дано 4 одномерных массива a,b,c,d длиной 10 элементов каждый. применить функцию для каждого из 4-х заданных массивов найти произведение найденных максимальных элементов. (в функции не должно быть операторов ввода или вывода)

26. Оформить функцию поиска **номера максимального** элемента массива среди элементов, стоящих на **нечетных** местах (*использовать шаг цикла $\neq 1$*)). В главной программе Дано 3 одномерных массива a,b,c длиной 30 элементов каждый. применить функцию для каждого из 3-х заданных массивов найти произведение найденных номеров максимальных элементов. (в функции не должно быть операторов ввода или вывода)

27. Оформить функцию поиска **произведения** положительных элементов массива, В главной программе Дано 4 одномерных массива a,b,c,d длиной 10 элементов каждый. применить функцию для каждого из 4-х заданных массивов. найти сумму найденных произведений. (в функции не должно быть операторов ввода или вывода)

28. Оформить функцию поиска **номера максимального** элемента массива среди отрицательных элементов, В главной программе Дано 3 одномерных массива a,b,c длиной 30 элементов каждый. применить функцию для каждого из 3-х заданных массивов. найти произведение найденных номеров максимальных элементов. (в функции не должно быть операторов ввода или вывода)

29. Оформить функцию поиска **номера минимального** элемента среди положительных элементов массива, В главной программе Дано 2 одномерных массива a,b длиной 30 элементов каждый. применить функцию для каждого из 2-х заданных массивов. найти разность найденных номеров минимальных элементов. (в функции не должно быть операторов ввода или вывода)

30. Оформить функцию поиска **суммы** положительных элементов массива, В главной программе Дано 4 одномерных массива a,b,c,d длиной 10 элементов каждый, применить функцию для каждого из 4-х заданных массивов. найти произведение найденных сумм элементов. (в функции не должно быть операторов ввода или вывода)

31. Оформить функцию поиска **суммы** положительных элементов

массива, стоящих на четных местах ((использовать шаг цикла $\neq 1$)), В главной программе Дано 3 одномерных массива a,b,c длиной 30 элементов каждый, применить функцию для каждого из 3-х заданных массивов.. найти произведение найденных сумм элементов. (в функции не должно быть операторов ввода или вывода)

32. Оформить функцию поиска **произведение** положительных элементов массива, стоящих на четных местах ((использовать шаг цикла $\neq 1$)), В главной программе Дано 3 одномерных массива a,b,c длиной 30 элементов каждый. применить функцию для каждого из 3-х заданных массивов. найти произведение найденных произведений элементов. (в функции не должно быть операторов ввода или вывода)

33. Оформить функцию поиска **суммы** квадратов элементов массива, В главной программе Дано 4 одномерных массива a,b,c,d длиной 10 элементов каждый, применить функцию для каждого из 4-х заданных массивов. найти произведение найденных сумм элементов. (в функции не должно быть операторов ввода или вывода)

задача 3 «Функции с массивами»

1. Описать функции нахождения **min** в одномерном массиве и функцию нахождения **max** в одномерном массиве. Дано 3 одномерных массива a,b,c длиной 20 элементов каждый. Использовать функции для облегчения решения задачи. Вычислить

$$t = \begin{cases} \min(b_i) * \max(a_i + c_i), & \text{если } \min(a_i) < \max(b_i) \\ \frac{\min(b_i + c_i)}{\min(a_i)} + \max(a_i), & \text{иначе} \end{cases}$$

где **min(a_i)** – означает наименьший элемент из массива a.

max(a_i) – означает наибольший элемент из массива a.

2. Описать функции нахождения **min** в одномерном массиве и функцию нахождения **max** в одномерном массиве. Дано 3 одномерных массива x,y,d длиной 40 элементов каждый. Использовать функции для облегчения решения задачи. Вычислить

$$t = \begin{cases} \min(d_i) + \max(x_i * y_i), & \text{если } \min(x_i) < \max(y_i) \\ \frac{\min(d_i + x_i)}{\min(d_i)} + \max(x_i), & \text{иначе} \end{cases}$$

где **min(a_i)** – означает наименьший элемент из массива a.

max(a_i) – означает наибольший элемент из массива a.

3. Описать подпрограмму-функцию для вычисления суммы кубов элементов массива – $\sum_{i=1}^{40} y_i^3$ для облегчения решения задачи: Дано 3

одномерных массива x, y, d длиной 40 элементов каждый. Вычислить

$$u = \begin{cases} \sum_{i=1}^{40} x_i^3 - \sum_{i=1}^{40} d_i^3, & \text{если } \sum_{i=1}^{40} y_i^3 > 0 \\ \sum_{i=1}^{40} y_i^3 / \sum_{i=1}^{40} d_i^3, & \text{иначе} \end{cases}$$

где $\sum_{i=1}^{40} y_i^3 = y_1^3 + y_2^3 + y_3^3 + \dots + y_{40}^3$ - означает сумма кубов элементов массива y

4. . Описать подходящую подпрограмму-функцию вычисления суммы элементов массива $\sum_{i=1}^{10} y_i$, для облегчения решения задачи: Дано 3 одномерных массива a, b, d длиной 10 элементов каждый. Вычислить

$$v = \begin{cases} \sum_{i=1}^{10} a_i - \sum_{i=1}^{10} d_i, & \text{если } \sum_{i=1}^{10} d_i > 0 \\ \sum_{i=1}^{10} d_i / \sum_{i=1}^{10} b_i, & \text{иначе} \end{cases}$$

где $\sum_{i=1}^{10} y_i = y_1 + y_2 + y_3 + \dots + y_{10}$ - означает сумма элементов массива y

5. . Описать подходящую подпрограмму-функцию (вычисления суммы элементов массива — $\sum_{i=1}^{10} y_i$) для облегчения решения задачи: Дано 3 одномерных массива a, b, d длиной 10 элементов каждый. Вычислить

$$v = \begin{cases} (\sum_{i=1}^{10} a_i + 1) / \sum_{i=1}^{10} d_i, & \text{если } \sum_{i=1}^{10} d_i > 0 \\ (\sum_{i=1}^{10} a_i + \sum_{i=1}^{10} b_i) \sum_{i=1}^{10} b_i, & \text{иначе} \end{cases}$$

где $\sum_{i=1}^{10} y_i = y_1 + y_2 + y_3 + \dots + y_{10}$ - означает сумма элементов массива y

6. . Описать подпрограмму-функцию вычисления произведения элементов массива — $\prod_{i=1}^{20} a_i$ для облегчения решения задачи. Дано 3 одномерных массива a, b, d длиной 20 элементов каждый. Вычислить

$$v = \begin{cases} \prod_{i=1}^{20} a_i - \prod_{i=1}^{20} (b_i + d_i), & \text{если } \prod_{i=1}^{20} a_i > \prod_{i=1}^{20} d_i \\ \prod_{i=1}^{20} (b_i - a_i) + \prod_{i=1}^{20} d_i, & \text{Иначе} \end{cases}$$

где $\prod_{i=1}^{20} a_i = a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot a_{20}$ - означает произведение элементов массива a

7. . Описать подходящую подпрограмму-функцию (вычисления суммы произведений элемента массива на число в степени —

$a_1x^{10} + a_2x^9 + a_3x^8 + \dots + a_9x + a_{10}$) . . Дано 2 одномерных массива а,в длиной 10 элементов каждый и 2 числа х,у. Вычислить выражение, вызвав 3 раза созданную функцию

$$\frac{(a_1x^{10} + a_2x^9 + a_3x^8 + \dots + a_9x + a_{10})^2 - (b_1y^{10} + b_2y^9 + b_3y^8 + \dots + b_9y + b_{10})}{b_1(x+y)^{10} + b_2(x+y)^9 + \dots + b_{10}}$$

8. . Описать подходящую подпрограмму-функцию (вычисления суммы произведений элемента массива на число в степени – $a_1x^{10} + a_2x^9 + a_3x^8 + \dots + a_9x + a_{10}$) для облегчения решения задачи. Дано 3 одномерных массива а,в,с длиной 10 элементов каждый и 3 числа х,у,з.. Вычислить выражение, вызвав 3 раза созданную функцию

$$\frac{(a_1x^{10} + a_2x^9 + a_3x^8 + \dots + a_9x + a_{10})^5 - (b_1y^{10} + b_2y^9 + b_3y^8 + \dots + b_9y + b_{10})^3}{(b_1(x+y)^{10} + b_2(x+y)^9 + \dots + b_{10})^2 + (a_1z^{10} + a_2z^9 + \dots + a_{10})}$$

9. .Дано 3 одномерных массива а,в,с длиной 10 элементов каждый и 3 числа х,у,з. Описать подходящую подпрограмму-функцию (вычисления суммы произведений элемента массива на число в степени) для облегчения решения задачи.

. Вычислить выражение, вызвав 3 раза созданную функцию $(a_1x^1 + a_2x^2 + a_3x^3 + \dots + a_9x^9 + a_{10}x^{10})^2 + (b_1y^1 + b_2y^2 + b_3y^3 + \dots + b_9y^9 + b_{10}y^{10})^3$

$$\frac{(b_1(x+y)^1 + b_2(x+y)^2 + \dots + b_{10}(x+y)^{10})^2 + (a_1z^1 + a_2z^2 + \dots + a_{10}z^{10})}{(b_1(x+y)^1 + b_2(x+y)^2 + \dots + b_{10}(x+y)^{10})^2 + (a_1z^1 + a_2z^2 + \dots + a_{10}z^{10})}$$

10 . Описать подпрограмму-функцию вычисления скалярного произведения векторов (массивов) $(\mathbf{x}, \mathbf{y}) = x_1y_1 + x_2y_2 + \dots + x_ny_n$ для облегчения решения задачи: Дано 3 одномерных массива (вектора) а,в,с длиной 10 элементов каждый и 3 числа х,у,з. Вычислить $(\mathbf{a}, \mathbf{b}) * \mathbf{x} + (\mathbf{b}, \mathbf{c}) * \mathbf{y} - (\mathbf{a}, \mathbf{c}) * ((\mathbf{a} + \mathbf{c}), \mathbf{b}) * \mathbf{z}$. (Где (\mathbf{a}, \mathbf{b}) – обозначает скалярное произведение векторов)

11. . Описать подпрограмму-функцию вычисления скалярного произведения векторов (массивов) $(\mathbf{x}, \mathbf{y}) = x_1y_1 + x_2y_2 + \dots + x_ny_n$ для облегчения решения задачи: Дано 3 одномерных массива (вектора) а,в,с длиной 20 элементов каждый и 3 числа х,у,з. Вычислить $(\mathbf{a}, \mathbf{c}) * \mathbf{z} + (\mathbf{a}, \mathbf{b}) * \mathbf{y} - (\mathbf{a}, (\mathbf{c} + \mathbf{b})) * ((\mathbf{a} - \mathbf{c}), \mathbf{b}) * \mathbf{z}$. (Где (\mathbf{a}, \mathbf{b}) – обозначает скалярное произведение векторов)

12. . Описать подходящую подпрограмму-функцию (вычисления суммы произведений элемента массива на число в степени) для облегчения решения задачи: Дано 3 одномерных массива а,в,с длиной 10 элементов каждый и 3 числа х,у,з.

. Вычислить выражение, вызвав 3 раза созданную функцию $(a_1x^1 + a_2x^2 + a_3x^3 + \dots + a_9x^9 + a_{10}x^{10})^2 + (b_1y^1 + b_2y^2 + b_3y^3 + \dots + b_9y^9 + b_{10}y^{10})^3$

$$\frac{(b_1(x+y)^1 + b_2(x+y)^2 + \dots + b_{10}(x+y)^{10})^2}{(b_1(x+y)^1 + b_2(x+y)^2 + \dots + b_{10}(x+y)^{10})^2}$$

13. . Описать подпрограмму-функцию вычисления произведения элементов массива – $\prod_{i=1}^{20} a_i$ для облегчения решения задачи: Дано 3

одномерных массива a, b, d длиной 20 элементов каждый. Вычислить

$$v = \begin{cases} \prod_{i=1}^{20} a_i - 5 \prod_{i=1}^{20} (b_i + d_i), & \text{если } \prod_{i=1}^{20} a_i > \prod_{i=1}^{20} d_i \\ 7 \prod_{i=1}^{20} (b_i - a_i) / \prod_{i=1}^{20} d_i, & \text{Иначе} \end{cases}$$

где $\prod_{i=1}^{20} a_i = a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot a_{20}$ - означает произведение элементов массива a

14. . Описать подпрограмму-функцию *вычисления суммы кубов элементов массива* – $\sum_{i=1}^{40} y_i^3$ для облегчения решения задачи. : Дано 3 одномерных массива x, y, d длиной 10 элементов каждый. Вычислить

$$u = \begin{cases} \sum_{i=1}^{10} x_i^3 + 3 \sum_{i=1}^{10} d_i^3, & \text{если } \sum_{i=1}^{10} y_i^3 > 0 \\ 8 \sum_{i=1}^{10} y_i^3 * \sum_{i=1}^{10} d_i^3, & \text{иначе} \end{cases}$$

где $\sum_{i=1}^{40} y_i^3 = y_1^3 + y_2^3 + y_3^3 + \dots + y_{40}^3$ - означает сумма кубов элементов массива

у

15. . Описать подходящую подпрограмму-функцию *вычисления суммы произведений элемента массива на число в степени* – $a_1 x^{10} + a_2 x^9 + a_3 x^8 + \dots + a_9 x + a_{10}$ для облегчения решения задачи: Дано 3 одномерных массива a, b, c длиной 10 элементов каждый и 3 числа x, y, z . . Вычислить выражение, вызвав 3 раза созданную функцию

$$\frac{(a_1 x^{10} + a_2 x^9 + a_3 x^8 + \dots + a_9 x + a_{10})^5 - (b_1 y^{10} + b_2 y^9 + b_3 y^8 + \dots + b_9 y + b_{10})^3}{(b_1 (x+y)^{10} + b_2 (x+y)^9 + \dots + b_{10})^3 + (a_1 z^{10} + a_2 z^9 + \dots + a_{10})}$$

Задача 4 «Текстовые строки и функции»

1. Описать функцию, вычисляющую количество появлений заданного символа в заданной строке («заданные» – это входные параметры функции). В главной программе дано 2 строки символов $S1$ и $S2$. Выяснить, что больше количество символов '*' в строке $S1$ или количество символов '+' в строке $S2$, используя функцию.

2. Описать функцию, вычисляющую количество всех слов в заданной строке. В главной программе дано 3 строки символов $S1$ и $S2$ и $S3$. Найти количество всех слов в этих строках, используя функцию.

3. Описать функцию, вычисляющую количество слов «Иванушка» в тексте. В главной программе дано 3 текста $S1$ и $S2$ и $S3$.

Выяснить, в каком тексте больше слов «Иванушка», используя функцию.

4. Описать функцию, вычисляющую количество цифр в тексте. В главной программе дано 2 строки символов S1 и S2. Выяснить, совпадает ли количество цифр в этих текстах, используя функцию.

5. Описать функцию, вычисляющую количество круглых скобок в тексте. В главной программе дано 2 текста S1 и S2. Выяснить, в каком тексте больше скобок, используя функцию.

6. . Описать функцию, вычисляющую количество появлений заданного символа в заданной строке («заданные» – это входные параметры функции).. В главной программе Дано 1 строка символов S1. Выяснить, совпадает ли количество круглых открывающихся скобок и круглых закрывающихся скобок в этом тексте, используя функцию.

7. Описать функцию, определяющую номер последней цифры в тексте. В главной программе дано 2 текста S1 и S2. Найти и вывести номера последних цифр в текстах.

8. Описать функцию, определяющую номер последнего символа равного заданному символу в заданном тексте («заданные» – это входные параметры функции).. В главной программе Дано 2 строки символов S1 и S2. Найти номер последнего символа «:» в S1 и номер последнего символа «;» в S2.

9. Описать функцию, вычисляющую количество латинских букв в тексте. В главной программе дано 2 текста S1 и S2. Выяснить, в каком тексте больше латинских букв, используя функцию.

10. . Описать функцию, вычисляющую количество появлений заданного символа в заданной строке («заданные» – это входные параметры функции).. В главной программе Дано 2 строки символов S1 и S2. Выяснить, что больше количество символов '@' в строке S1 или количество символов '\$' в строке S2, используя функцию.

11. Описать функцию, вычисляющую количество предложений в заданной строке. В главной программе дано 3 строки символов S1 и S2 и S3. Найти количество всех предложений в этих строках, используя функцию.

12. Описать функцию, вычисляющую количество квадратных скобок в тексте. В главной программе дано 2 текста S1 и S2. Выяснить, в каком тексте больше скобок, используя функцию.

13. . Описать функцию, определяющую номер последней

закрывающейся скобки (круглой или квадратной) в тексте. В главной программе дано 2 текста S1 и S2. Найти и вывести номера последних скобок в текстах.

14. . описать функцию, вычисляющую количество появлений заданного слова в заданной строке («заданные» – это входные параметры функции).. В главной программе дано 2 строки символов s1 и s2. выяснить, что больше количество появлений слова 'Pascal' в строке s1 или в строке s2, используя функцию.

15. . **Описать** функцию, вычисляющую количество появлений заданного символа в заданной строке («заданные» – это входные параметры функции).. В главной программе Дано 2 строки символов S1 и S2. Выяснить, что больше количество букв 'Z' в строке S1 или в строке S2, используя функцию.

16. . . Описать функцию, вычисляющую количество появлений заданного слова в заданной строке («заданные» – это входные параметры функции).. В главной программе Дано 1 строка символов S1. Выяснить, совпадает ли количество появлений слова 'Begin' в строке S1 и количество появлений слова 'End' , используя функцию.

17. . Описать функцию, вычисляющую количество появлений заданного слова в заданной строке («заданные» – это входные параметры функции).. В главной программе Дано 2 строки символов S1 и S2. Выяснить, что больше количество появлений слова 'Иванов' в строке S1 или количество появлений слова 'Петров' в строке S2. ., используя функцию.

18. . Описать функцию, вычисляющую количество появлений заданного символа в заданной строке («заданные» – это входные параметры функции).. В главной программе Дано 2 строки символов S1 и S2. Выяснить, что больше количество букв 'к' в строке S1 или количество букв 'н' в строке S2, используя функцию.

19. . Описать функцию, вычисляющую количество появлений заданного слова в заданной строке («заданные» – это входные параметры функции).. В главной программе Дано 2 строки символов S1 и S2. Выяснить, что больше количество появлений слова 'Информатика' в строке S1 или количество появлений слова 'Технология' в строке S2. ., используя функцию.

Задача 5 «функции и структурный тип»

1. Создать комбинированный (структурный) тип для сведений о периодических изданиях (наименование издания, тираж, годовая стоимость). Описать функцию нахождения общей суммы стоимостей изданий в одном таком комбинированном массиве. Пользователь задает два комбинированных

массива по N элементов в каждом.(для двух библиотек). Применить функцию два раза для заданных двух библиотек. (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

2. Создать комбинированный (структурный) тип для меню детского кафе (наименование изделия, вес, стоимость). Описать функцию нахождения наименования самого дорого блюда дня в одном таком комбинированном массиве. Пользователь задает два комбинированных массива по N элементов в каждом. (два меню для разных дней). Применить функцию два раза для заданных двух меню. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

3. Создать комбинированный (структурный) тип для меню ресторана "Дракон" (наименование изделия, вес, стоимость). Описать функцию нахождения общего веса изделий в одном таком комбинированном массиве. Пользователь задает два комбинированных массива по N элементов в каждом (для двух отделений ресторана). Применить функцию два раза для заданных двух отделений ресторана. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

4. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать функцию нахождения общей длительности всех музыки на всех дисках в коллекции. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

5. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать функцию нахождения количества дисков с указанным исполнителем в коллекции. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

6. Создать комбинированный (структурный) тип для анкетные данные студентов (Ф. И. О., год рождения, адрес, сведения о родителях, средний балл). Описать функцию нахождения лучшего студента в группе (по среднему баллу). Пользователь задает два комбинированных массива по N элементов в каждом.(для двух групп). Применить функцию два раза для заданных двух групп. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

7. Создать комбинированный (структурный) тип для расписание полетов самолетов (пункт посадки, время отправления, время прибытия, время полета, стоимость билета). Описать функцию нахождения самого

короткого полета в одном таком комбинированном массиве. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух аэропортов). Применить функцию два раза для заданных двух аэропортов. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

8. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать функцию нахождения самого дорого диска в коллекции. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

9. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать функцию нахождения количества дисков не старше указанного года в коллекции. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

10. Создать комбинированный (структурный) тип для перечень товаров, имеющих в продаже в магазине "Океан" (наименование, единица измерения, цена, количество). Описать функцию нахождения общей суммы стоимостей (цена*количество) в одном таком комбинированном массиве. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух магазинов). Применить функцию два раза для заданных двух магазинов. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

11. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать функцию нахождения количества дисков с указанным годом в коллекции. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

12. Создать комбинированный (структурный) тип для график отпусков (Ф. И. О., дата начала отпуска, дата выхода на работу, количество дней). Описать функцию нахождения самого короткого отпуска в одном таком комбинированном массиве. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух организаций). Применить функцию два раза для заданных двух организаций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

13. Создать комбинированный (структурный) тип для ин-

формацию о тестируемом (Ф.И.О., IQ, возраст). Описать функцию нахождения фамилии самого умного в группе. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух групп). Применить функцию два раза для заданных двух групп. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

14. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать функцию нахождения количества дисков с указанным стилем в коллекции. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

15. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать функцию нахождения общей суммы стоимостей дисков в коллекции. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

Дополнительные* задачи

задача 4*

1. **задача 4.** *Описать функцию **DigitN(K,N)** целого типа, возвращающую N -ю цифру целого положительного числа K (цифры в числе нумеруются справа налево). Если количество цифр в числе K меньше N , то функция возвращает **-1**. Для каждого из пяти данных целых положительных чисел $K1, K2, \dots, K5$ вызвать функцию **DigitN** с параметром N , изменяющимся от **1** до **5**.

2. **задача 4** Описать функцию **DigitCount(K)** целого типа, находящую количество цифр целого положительного числа K . Используя эту функцию, найти количество цифр для каждого из пяти данных целых положительных чисел.

3. **Задача 4.** *Даны натуральные числа a, b, c . Определить функцию **bin(x)**, переводящую число x из десятичной системы счисления в двоичную. Найти **bin(a + b)**, **bin(ab + c)**.

4. **Задача 4.** Даны два натуральных числа a, b . Вычислить $\frac{a!! - ab}{a!! + ab}$.
Функция $x!!$ определяется следующим образом: $x!! = 1*3*5*...*x$, если x нечетно, $x!! = 2*4*6*...*x$, если x четно.

5. **Задача4.** Описать функцию **Fact2(N)** вещественного типа,

вычисляющую *двойной факториал*: $N!! = 1 * 3 * 5 * \dots * N$, если N — нечетное; $N!! = 2 * 4 * 6 * \dots * N$, если N — четное ($N > 0$ — параметр целого типа; вещественное возвращаемое значение используется для того, чтобы избежать целочисленного переполнения при больших значениях N). С помощью этой функции найти двойные факториалы пяти данных целых чисел.

6. **задача 4.** Описать функцию $\text{Fib}(N)$ целого типа, вычисляющую N -й элемент последовательности *чисел Фибоначчи* F_k , которая описывается следующими формулами: $F_1 = 1$, $F_2 = 1$, $F_k = F_{k-2} + F_{k-1}$, $k = 3, 4, \dots$. Используя функцию Fib , найти пять чисел Фибоначчи с данными номерами N_1 .

7. **задача 4.** Даны три квадратные матрицы третьего порядка. Вывести на экран ту из них, норма которой наименьшая. В качестве нормы матрицы взять максимум модулей ее элементов. Нахождение нормы матрицы оформить в виде функции.

8. **задача 4.** Заданы матрицы A и B . Переменной S присвоить -1 , если максимальный элемент матрицы A больше максимального элемента матрицы B ; 0 если максимальные элементы матрицы равны; 1 , если максимальный элемент матрицы A меньше максимального элемента матрицы B . Поиск максимального элемента оформить в виде функции.

9. **задача 4.** Заданы три матрицы A (третьего порядка), B (второго порядка) и C (третьего порядка). Найти максимальное из трех чисел x , y , z , где x - след матрицы A , y - след матрицы B , z - след матрицы C . (Следом матрицы называется сумма элементов главной диагонали). Вычисление следа матрицы оформить в виде функции.

10. **задача 4.** Заданы два вектора $X(x_1, x_2, x_3)$, $Y(y_1, y_2, y_3)$ и матрица A третьего порядка. Найти сумму двух векторов c и d , где вектор c есть произведение вектора x на матрицу A , а вектор d - произведение вектора на матрицу A . Вычисление произведения вектора на матрицу оформить в виде процедуры.

11. **задача 4.** *Описать функцию $\text{DigitN}(K, N)$ целого типа, возвращающую N -ю цифру целого положительного числа K (цифры в числе нумеруются справа налево). Если количество цифр в числе K меньше N , то функция возвращает -1 . Для каждого из пяти данных целых положительных чисел K_1, K_2, \dots, K_5 вызвать функцию DigitN с параметром N , изменяющимся от 1 до 5 .

12. **Задача 4.** *Даны натуральные числа a, b, c . Определить функцию $\text{bin}(x)$, переводящую число x из десятичной системы счисления в двоичную. Найти $\text{bin}(a + b)$, $\text{bin}(ab + c)$.

13. **Задача 4.** Даны две квадратные матрицы A, B . Построить таблицу функции $y = cx^2 + d$ при x меняющемся от 0 до 1 с шагом 0.1 , где $c = \text{sp}(A)$, $d = \text{sp}(B)$. ($\text{sp}(A)$ - след матрицы A - сумма элементов главной диагонали).

14. **задача 4.** Описать подходящую подпрограмму-функцию для облегчения решения задачи. Вычислить $C_n^m = \frac{n!}{m!(n-m)!}$

15. задача 4 Описать функцию SumRange(A, B) целого типа, находящую сумму всех целых чисел от A до B включительно (A и B — целые). Если $A > B$, то функция возвращает 0. С помощью этой функции найти суммы чисел от A до B и от B до C , если даны числа A, B, C .

16. задача 4 Описать подходящую подпрограмму-функцию для облегчения решения задачи. Вычислить $A_n^m = \frac{n!}{(n-m)!}$

ФУНКЦИИ, НЕ ВОЗВРАЩАЮЩИЕ ЗНАЧЕНИЯ

Эти функции объявляются типом **void** и не имеют в своем теле оператора **return()**; Вызываются в отдельном операторе, *начинающемся с имени функции*.

```
void Имя_функции (список_параметров_и_их_типов)
```

```
{
Тело_функции
}
```

Пример

```
void vyvod (int x, int y)
```

```
{cout<<" первое число="<<x<<"второе число="<<y<<endl;}
```

```
int main()
```

```
int a,b,c,y;
```

```
cin>>a>>b>>c>>y;
```

```
Vyvod(a,b);// вызов функции типа void
```

Vyvod(y,c);//вызов функции типа **void** в отдельном операторе начинающемся с имени функции

```
}
```

Такие функции Могут возвращать в главную программу несколько вычисленных значений, в отличие от функций, в которых выходной результат помещается всегда в операторе **return()**; и может быть только в единственном числе, тип: **int** , **char**, **boolean**.

ПЕРЕДАЧА ПАРАМЕТРОВ ФУНКЦИИ

Передача параметров функции происходит одним из трех способов:

- по значению; **int** x;
- по указателю на переменную.**int** *x;
- по ссылке; **int** &x; (по адресу переменной)

ПЕРЕДАЧА ПАРАМЕТРОВ ПО ЗНАЧЕНИЮ

Рассмотрим различие этих способов на одном примере – на примере функции **change()** занимающейся перестановкой (обменом) двух

переменных. Первый случай *передачи параметров по значению* на примере функции, обменивающей значения двух переменных между собой:

Пример:

```
void change (int x, int y)
{int temp = x; x = y; y = temp;
cout<< “Внутри функции x = “ << x;
cout<< “Внутри функции y= “ <<y;
}
void main()
{int a = 3, b = 5;
cout << “до вызова функции a = “ << a; // до вызова функции a = 3
cout<< “до вызова функции b = “ << b ; //до вызова функции b = 5
change (a,b); /*внутри функции a = x = 5, b = y = 3, значения
изменились*/
cout<< “после вызова функции a = ” << a; /* a = 5, значение не
изменилось*/
cout<< “ после вызова функции b = “ << b ; /* b=3 значение не
изменилось*/
}
```

В этом примере при вызове функции **change** формальному параметру **x** присваивается значение фактического параметра **a**, равного **3**. Так как **x** является локальной переменной функции **change**, то ее значение доступно только внутри функции. Если значение **x** будет изменено в функции, то новое ее значение не будет доступно в главной программе, то есть не повлияет на значение **a**.

Если необходимо, что бы изменённое значение более двух переменных было доступно в главной программе невозможно использовать оператор **return**, так как оператор **return** не может возвращать более одного значения, то для организации *функции, возвращающей более одного* изменённого значения, используются способы передачи аргументов *по ссылке или по указателю*. Для этого формальный параметр функции описывается как *указатель*, а при вызове функции в главной программе в качестве фактического параметра указывается адрес переменной, получаемый с помощью *операции получения адреса* с символом **&** «амперсанд» (см. гл. «4.1 Указатели, адреса и ссылки»)

ПЕРЕДАЧА ПАРАМЕТРОВ ПО УКАЗАТЕЛЮ

Рассмотрим второй случай *передачи параметров по указателю* на примере этой же функции обменивающей значения двух переменных, но будем уже использовать указатели на переменные.

Пример:

```
void change (int *x, int *y)
{int temp = *x; *x = *y; *y = temp;
cout<< “Внутри функции x = “ << *x;
```

```

cout<< “Внутри функции y= “ <<*y;
}
void main()
{int a = 3, b = 5;
cout << “до вызова функции a = “ << a; // до вызова функции a = 3
cout<< “до вызова функции b = “ << b ; //до вызова функции b = 5
change (&a, &b); // внутри функции a = x = 5, b = y = 3
cout<< “после вызова функции a = ” << a; //после вызова функции
a=5
cout<<“ после вызова функции b = “ << b ;//после вызова функции
b=3
}

```

При вызове функции указателю *x присваивается адрес переменной a (x = &a), а не значение переменной как в первом случае. Внутри функции происходит запись нового значения не в локальную переменную, а по указанному адресу внешней переменной a, что приводит к изменению значения переменной a, в главной функции.

Задания для лабораторной работы №3. Особенности создания функций, не возвращающих значения

задача № 1 «Линейные функции типа void»

1) Создать функцию типа void с передачей параметров по ссылке или указателю, которая переводит время, заданное в минутах в секунды. . Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

2) Создать функцию типа void с передачей параметров по ссылке или указателю, которая определяет периметр треугольника по трем его сторонам. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

3) Создать функцию типа void с передачей параметров по ссылке или указателю, которая возвращает номер квадранта, в котором находится точка. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

4) Создать функцию типа void с передачей параметров по ссылке

или указателю, которая возвращает среднее арифметическое трех данных чисел. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

5) Создать функцию типа void с передачей параметров по ссылке или указателю, которая определяет площадь круга по его радиусу. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

6) Создать функцию типа void с передачей параметров по ссылке или указателю, которая возвращает остаток от деления двух натуральных чисел. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

7) Создать функцию типа void с передачей параметров по ссылке или указателю, которая переводит радианы в градусы. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

8) Создать функцию типа void с передачей параметров по ссылке или указателю, которая определяет длину отрезка по его координатам. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

9) Создать функцию типа void с передачей параметров по ссылке или указателю, которая возвращает в долларах сумму, заданную в рублях. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

10) Создать функцию типа void с передачей параметров по ссылке или указателю, которая возвращает большее из двух данных чисел. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать

созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

11) Создать функцию типа void с передачей параметров по ссылке или указателю, которая определяет длину окружности по заданному радиусу. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

12) Создать функцию типа void с передачей параметров по ссылке или указателю, которая переводит скорость из км/час в м/сек. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

13) Создать функцию типа void с передачей параметров по ссылке или указателю, которая возвращает среднее геометрическое двух данных чисел. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

14) Создать функцию типа void с передачей параметров по ссылке или указателю, которая возвращает в рублях сумму, заданную в долларах. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

15) Создать функцию типа void с передачей параметров по ссылке или указателю, которая переводит градусы в радианы. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

16) Создать функцию типа void с передачей параметров по ссылке или указателю, которая возвращает меньшее из двух данных чисел. Для создаваемой функции: подобрать имя; выбрать имена и типы входных и выходных параметров; описать тело функции; в главной программе вызвать созданную подпрограмму более одного раза с различными входными данными. Вывести результаты в главной программе.

17) Создать функцию типа void с передачей параметров по ссылке

или указателю, которая переводит время, заданное в секундах в минуты

задача № 2

Вариант № 1 Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики параллелограмма по заданным двум сторонам и углу между ними. Выходные результаты функции – характеристики параллелограмма – это периметр и площадь $s = a b \cos \alpha$. В главной программе задано два параллелограмма. Найти их характеристики, вызвав функцию 2 раза.

Вариант №2 Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики квадрата по заданной стороне. Выходные результаты функции – Характеристики квадрата – это периметр, диагональ и площадь. В главной программе задано два квадрата. Найти их характеристики, вызвав функцию 2 раза.

Вариант №3 Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики прямоугольника по заданным двум сторонам . Выходные результаты функции – Характеристики прямоугольника – это периметр, диагональ и площадь . В главной программе задано два прямоугольника . Найти их характеристики, вызвав функцию 2 раза.

Вариант № 4 Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики равнобедренной трапеции по заданным двум основаниям a,b и высоте h. Выходные результаты функции Характеристики равнобедренной трапеции – это периметр, и площадь $s = \frac{1}{2} (a + b) h$. В главной программе задано две равнобедренные трапеции . Найти их характеристики, вызвав функцию 2 раза.

Вариант № 5. Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики окружности по заданному радиусу R. Выходные результаты функции Характеристики окружности – это длина окружности и площадь . В главной программе задано две окружности . Найти их характеристики, вызвав функцию 2 раза.

Вариант № 6. Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики сферы по заданному радиусу R. Выходные результаты функции –

характеристики сферы – это площадь поверхности, $s=4 \pi R^2$ и объем сферы $v=4/3 \pi R^3$. В главной программе задано две сферы. Найти их характеристики, вызвав функцию 2 раза.

Вариант № 7. Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики квадратной призмы по заданной высоте H и стороне основания a. Выходные результаты функции – характеристики квадратной призмы – это площадь основания и объем. В главной программе задано две квадратные призмы. Найти их характеристики, вызвав функцию 2 раза.

Вариант № 8. Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики правильной треугольной призмы по заданной высоте H и стороне основания a. Выходные результаты функции – характеристики правильной треугольной призмы – это площадь основания и объем. В главной программе задано две треугольные призмы. Найти их характеристики, вызвав функцию 2 раза.

Вариант №9 Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики цилиндра по заданной высоте H и радиусу основания R. Выходные результаты функции – характеристики цилиндра – это площадь основания и объем. В главной программе задано два цилиндра. Найти их характеристики, вызвав функцию 2 раза.

Вариант №10 Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики цилиндра по заданной высоте H и радиусу основания R. Выходные результаты функции – характеристики цилиндра – это площадь боковой поверхности и объем. В главной программе задано два цилиндра. Найти их характеристики, вызвав функцию 2 раза.

Вариант №11 Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики конуса по заданной высоте H и радиусу основания R. Выходные результаты функции – характеристики конуса – это площадь боковой поверхности $S=\pi R \sqrt{R^2 + H^2}$ и объем $v=1/3 H \pi R^2$. В главной программе задано два конуса. Найти их характеристики, вызвав функцию 2 раза.

Вариант №12 Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики конуса по заданной высоте H и радиусу основания R. Выходные результаты

функции – *характеристики конуса* – это площадь основания и объем $v = 1/3 \pi R^2 H$. В главной программе задано два конуса. Найти их характеристики, вызвав функцию 2 раза.

Вариант №13. Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики Усеченного конуса по заданной высоте H и двум радиусам оснований R и r. Выходные результаты функции – характеристики Усеченного конуса – это площадь нижнего основания и объем $v = 1/3 \pi (R^2 + r^2 + Rr)H$. В главной программе задано два конуса. Найти их характеристики, вызвав функцию 2 раза.

Вариант №14 Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики вектора на плоскости по заданным координатам x и y. Выходные результаты функции – характеристики вектора – это длина вектора и тангенс угла наклона (между осью OX и вектором). В главной программе задано два вектора. Найти их характеристики, вызвав функцию 2 раза.

Вариант №15. Реализовать функцию типа void с передачей параметров-результатов по ссылке. Функция вычисляет характеристики точки на плоскости по заданным координатам x и y. Выходные результаты функции – характеристики точки – это расстояние до начала координат и номер четверти, в которой находится точка. В главной программе задано две точки. Найти их характеристики, вызвав функцию 2 раза.

Задача № 3. функции типа void обрабатывающие тексты

1. . Описать функцию типа void с передачей параметров по ссылке или указателю POWER3(A,B), вычисляющую третью степень числа A и возвращающую ее в переменной B (A — входной, B — выходной параметр; оба параметра являются вещественными). С помощью этой функции найти третьи степени пяти данных чисел.

2. Описать функцию типа void с передачей параметров по ссылке или указателю, заменяющую в тексте слова «обязательно» на «возможно». В главной программе Дано 2 строки символов S1 и S2.. Применить функцию в главной программе для строк S1 и S2..

3. . Описать функцию типа void с передачей параметров по ссылке или указателю ShiftLeft(A, B, C), выполняющую *левый циклический сдвиг*: значение A переходит в C, значение C — в B, значение B — в A (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой функции в главной программе выполнить левый циклический сдвиг для двух данных наборов из трех чисел: (A1, B1, C1) и (A2, B2, C2),

4. Описать функцию типа void с передачей параметров по ссылке или указателю, заменяющую в тексте все малые буквы после точки и пробела

на большие . В главной программе Дано 2 строки символов S1 и S2. Применить функцию для строк S1 и S2.

5. . Описать функцию типа void с передачей параметров по ссылке или указателю POWERA234(A,B,C,D), вычисляющую вторую, третью и четвертую степень числа A и возвращающую эти степени соответственно в переменных B, C, D (A — входной, B, C, D — выходные параметры; все параметры являются вещественными). С помощью этой функции в главной программе найти вторую, третью и четвертую степень пяти данных чисел.

6. . Описать функцию типа void с передачей параметров по ссылке или указателю, заменяющую в тексте слова «Pascal» на «C++». В главной программе Дано 3 строки символов S1 и S2, S3. Применить функцию в главной программе для строк S1 и S2, S3.

7. Описать функцию типа void с передачей параметров по ссылке или указателю, заменяющую в тексте указанное слово на другое заданное слово . В главной программе Дано 2 строки символов S1 и S2. В тексте S1 заменить слово «Пиноккио» на «Буратино», а в тексте S2 заменить слово «Мальвина» на «Марина», применяя функцию

8. . Описать функцию типа void с передачей параметров по ссылке или указателю $Mean(X, Y, AMean, GMean)$, вычисляющую *среднее арифметическое* $AMean = (X + Y)/2$ и *среднее геометрическое* $GMean = \sqrt{X*Y}$ двух положительных чисел X и Y (X и Y — входные, $AMean$ и $GMean$ — выходные параметры вещественного типа). С помощью этой функции в главной программе найти среднее арифметическое и среднее геометрическое для пар $(A, B), (A, C), (A, D)$, если даны A, B, C, D .

9. . Описать функцию типа void с передачей параметров по ссылке или указателю, заменяющую в тексте указанное слово на другое заданное слово . В главной программе Дано 2 строки символов S1 и S2. В тексте S1 заменить слово «Грэтель» на «Василиса». В тексте S2 заменить слово «Синдерелла» на «Золушка», применяя функцию

10. . Описать функцию типа void с передачей параметров по ссылке или указателю $TrianglePS(a, P, S)$, вычисляющую по стороне a равностороннего треугольника его периметр $P = 3*a$ и площадь $S = a^2*\sqrt{3}/4$ (a — входной, P и S — выходные параметры; все параметры являются вещественными). С помощью этой функции в главной программе найти периметры и площади трех равносторонних треугольников с данными сторонами.

11. . Описать функцию типа void с передачей параметров по ссылке или указателю, заменяющую в тексте все малые буквы на большие . В главной программе Дано 2 строки символов S1 и S2. Применить функцию в главной программе для строк S1 и S2.

12. . Описать функцию типа void с передачей параметров по ссылке или указателю $RectPS(x1, y1, x2, y2, P, S)$, вычисляющую периметр P и площадь S прямоугольника со сторонами, параллельными осям координат,

по координатам $(x1, y1)$, $(x2, y2)$ его противоположных вершин $(x1, y1, X2, Y2$ — входные, P и S — выходные параметры вещественного типа). С помощью этой функции в главной программе найти периметры и площади трех прямоугольников сданными противоположными вершинами,

13. . Описать функцию типа void с передачей параметров по ссылке или указателю, заменяющую в тексте все большие буквы на малые . В главной программе Дано 3 строки символов $S1$ и $S2$, $S3$.. Применить функцию в главной программе для строк $S1$ и $S2$, $S3$.

14. . Описать функцию типа void с передачей параметров по ссылке или указателю $\text{Swap}(X, Y)$, меняющую содержимое переменных X и Y (X и Y — вещественные параметры, являющиеся одновременно входными и выходными). С ее помощью в главной программе для данных переменных A, B, C, D последовательно поменять содержимое следующих пар: A и B , C и D , B и C и вывести новые значения A, B, C, D .

15. . Описать функцию типа void с передачей параметров по ссылке или указателю $\text{Minmax}(X, Y)$, записывающую в переменную X минимальное из значений X и Y , а в переменную Y — максимальное из этих значений (X и Y — вещественные параметры, являющиеся одновременно входными и выходными). Используя четыре вызова этой функции в главной программе , найти минимальное и максимальное из данных чисел A, B, C, D .

16. . Описать функцию типа void с передачей параметров по ссылке или указателю $\text{ShiftRight3}(A, B, C)$, выполняющую *правый циклический сдвиг*: значение A переходит в B , значение B — в C , значение C — в A (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой функции в главной программе выполнить правый циклический сдвиг для двух данных наборов из трех чисел: $(A1, B1, C1)$ и $(A2, B2, C2)$.

Задача 4. Подпрограммы-процедуры или функции типа void.

3. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, переводящую декартовы координаты заданной точки (x, y) в полярные координаты по следующим формулам:

$$\rho = \sqrt{x^2 + y^2}; \quad \varphi = \arctg\left(\frac{y}{x}\right);$$
Входные параметры: (x, y) , выходные параметры (по ссылке или указателю): ρ, φ .
 В главной программе для двух заданных точек $(x1, y1)$ и $(x2, y2)$ найти их полярные координаты и найти сумму их углов $\varphi_1 + \varphi_2$ и разность радиус векторов $\rho^1 - \rho^2$.

4. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, переводящую декартовы координаты заданной точки (x, y, z) в цилиндрические координаты по следующим формулам:

$\rho = \sqrt{x^2 + y^2}; \quad \varphi = \arctg\left(\frac{y}{x}\right); \quad z_2 = z$ *Входные параметры:* (x, y, z) , *выходные*

параметры (по ссылке или указателю): ρ, φ, z . В главной программе для двух заданных точек найти их цилиндрические координаты и найти сумму их углов $\varphi_1 + \varphi_2$ и разность радиус векторов $\rho^1 - \rho^2$.

5. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, переводящую декартовы координаты заданной точки (x, y, z) в полярные координаты по следующим формулам:

$$\rho = \sqrt{x^2 + y^2 + z^2}; \quad \varphi = \arctg\left(\frac{y}{x}\right);$$

$$z = \arctg\left(\frac{\sqrt{x^2 + y^2}}{z}\right)$$

Входные параметры: (x, y, z) , *выходные*

параметры (по ссылке или указателю): ρ, φ, z . В главной программе для двух заданных точек найти их полярные координаты и найти сумму их углов $\varphi_1 + \varphi_2$ и разность радиус векторов $\rho^1 - \rho^2$.

6. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, определяющую середину отрезка (точку $O(x_0, y_0)$), заданного координатами его концов (точки $M_1(x_1, y_1)$ $M_2(x_2, y_2)$) следующим формулам:

$$x_0 = \frac{x_1 + x_2}{2}; \quad y_0 = \frac{y_1 + y_2}{2}$$

Входные параметры: x_1, y_1, x_2, y_2 , *выходные параметры (по ссылке или указателю):* x_0, y_0 .

В главной программе для двух заданных отрезков выяснить, не совпадают ли их середины.

7. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, определяющую точку $P(x_p, y_p)$, делящую заданный отрезок в отношении $\frac{1}{\lambda}$ (отрезок, задан координатами его концов (точки $M_1(x_1, y_1)$ $M_2(x_2, y_2)$) по следующим формулам:

$$x_p = \frac{x_1 + \lambda \cdot x_2}{1 + \lambda}; \quad y_p = \frac{y_1 + \lambda \cdot y_2}{1 + \lambda}$$

Входные параметры: $x_1, y_1, x_2, y_2, \lambda$, *выходные параметры (по ссылке или указателю):* x_p, y_p . В главной программе для двух заданных отрезков выяснить не совпадают ли их вычисленные полюса.

8. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, определяющую координаты центра тяжести (точку $O(x_0, y_0)$), системы материальных частиц с координатами $M_i(x_i, y_i)$ и с массами m_i . (должны быть заданы три массива x_i, y_i, m_i для задания одной системы частиц) по следующим формулам:

$$x_0 = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i}; \quad y_0 = \frac{\sum_{i=1}^n m_i y_i}{\sum_{i=1}^n m_i}; \quad \text{Входные параметры: три массива } x_i, y_i, m_i,$$

выходные параметры (по ссылке или указателю): x_0, y_0 .
В главной программе для двух заданных систем материальных частиц выяснить не совпадают ли их центры тяжести.

9. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, заполняющую одномерный массив $x(n)$ элементами побочной диагонали матрицы $A(n*n)$. *Входные параметры:* матрица $A(n*n)$, *выходные параметры (по ссылке или указателю):* массив $x(n)$. В главной программе для двух заданных матриц $C1(n*n)$ и $C1(m*m)$ создать такие массивы и проверить равны ли их серединные элементы (т.е. элементы, расположенные в центре массива, если кол-во четное ,то ,на ваше усмотрение, левый или правый элемент от центра)

10. Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, переводящую декартовы координаты заданной точки (x,y) в полярные координаты по следующим формулам:
 $\rho = \sqrt{x^2 + y^2}; \quad \varphi = \arctg\left(\frac{y}{x}\right);$ *Входные параметры:* (x,y) , *выходные параметры (по ссылке или указателю):* ρ, φ .
В главной программе для двух заданных точек найти их полярные координаты и найти сумму их углов $\varphi_1 + \varphi_2$ и разность радиус векторов $\rho_1 - \rho_2$.

11. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, переводящую декартовы координаты заданной точки (x,y,z) в цилиндрические координаты по следующим формулам:

$\rho = \sqrt{x^2 + y^2}; \quad \varphi = \arctg\left(\frac{y}{x}\right); \quad z_2 = z.$ *Входные параметры:* (x,y,z) , *выходные параметры (по ссылке или указателю):* ρ, φ, z_2 . В главной программе для двух заданных точек найти их цилиндрические координаты и найти сумму их углов $\varphi_1 + \varphi_2$ и разность радиус векторов $\rho_1 - \rho_2$.

12. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, переводящую декартовы координаты заданной точки (x,y,z) в полярные координаты по следующим формулам:

$\rho = \sqrt{x^2 + y^2 + z^2}; \quad \varphi = \arctg\left(\frac{y}{x}\right);$
 $z = \arctg\left(\frac{\sqrt{x^2 + y^2}}{z}\right)$. *Входные параметры:* (x,y,z) , *выходные параметры (по ссылке или указателю):* ρ, φ, z .

В главной программе для

двух заданных точек найти их полярные координаты и найти сумму их углов $\varphi_1 + \varphi_2$ и разность радиус векторов $\rho_1 - \rho_2$.

13. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, определяющую середину отрезка (точку $O(x_0, y_0)$), заданного координатами его концов (точки $M_1(x_1, y_1)$ $M_2(x_2, y_2)$) следующим формулам:

$$x_0 = \frac{x_1 + x_2}{2}; \quad y_0 = \frac{y_1 + y_2}{2}$$
Входные параметры: x_1, y_1, x_2, y_2 , *выходные параметры (по ссылке или указателю):* x_0, y_0 .
 В главной программе для двух заданных отрезков выяснить, не совпадают ли их середины.

14. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, определяющую точку $P(x_p, y_p)$, делящую заданный отрезок в отношении $\frac{1}{\lambda}$ (отрезок, задан координатами его концов (точки $M_1(x_1, y_1)$ $M_2(x_2, y_2)$) по следующим формулам:

$$x_p = \frac{x_1 + \lambda x_2}{1 + \lambda}; \quad y_p = \frac{y_1 + \lambda y_2}{1 + \lambda}$$
Входные параметры: $x_1, y_1, x_2, y_2, \lambda$, *выходные параметры (по ссылке или указателю):* x_p, y_p . В главной программе для двух заданных отрезков выяснить не совпадают ли их вычисленные полюса.

15. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, определяющую координаты центра тяжести (точку $O(x_0, y_0)$), системы материальных частиц с координатами $M_i(x_i, y_i)$ и с массами m_i . (должны быть заданы три массива x_i , y_i , m_i для задания одной системы частиц) по следующим формулам:

$$x_0 = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i}; \quad y_0 = \frac{\sum_{i=1}^n m_i y_i}{\sum_{i=1}^n m_i}; \quad \text{Входные параметры: три массива } x_i, y_i, m_i,$$

выходные параметры (по ссылке или указателю): x_0, y_0 .
 В главной программе для двух заданных систем материальных частиц выяснить не совпадают ли их центры тяжести.

16. . Описать подпрограмму-функцию типа void с передачей параметров по ссылке или указателю, заполняющую одномерный массив $x(n)$ элементами побочной диагонали матрицы $A(n \times n)$. В главной программе для двух заданных матриц $C_1(n \times n)$ и $C_1(m \times m)$ создать такие массивы и проверить равны ли их срединные элементы (т.е. элементы, расположенные в центре массива, если кол-во четное ,то ,на ваше усмотрение, левый или правый элемент от центра)

Задача 5 «Функции и процедуры . Квадрат матрицы».

1. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C . Вычислить квадрат той матрицы ($A * A$), в которой след наименьший. Вычисление следа оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры. (След матрицы – это сумма элементов главной диагонали)

2. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C . Вычислить квадрат той матрицы ($A * A$), в которой сумма элементов побочной диагонали наибольшая. Вычисление суммы элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

3. **Задача 4 «Функции+ процедура и».** Дано 3 квадратные матрицы A, B и C . Вычислить квадрат той матрицы ($A * A$), в которой сумма элементов первого столбца наибольшая. Вычисление суммы элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

4. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C . Вычислить квадрат той матрицы ($A * A$), в которой сумма элементов последней строки наибольшая. Вычисление суммы элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

5. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C . Вычислить квадрат той матрицы ($A * A$), в которой сумма элементов третьей строки наименьшая. Вычисление суммы элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

6. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C . Вычислить квадрат той матрицы ($A * A$), в которой сумма элементов третьей строки наименьшая. Вычисление квадрата матрицы оформить в виде процедуры. Вычисление квадрата матрицы оформить в виде процедуры

7. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C . Вычислить квадрат той матрицы ($A * A$), в которой количество нулевых элементов третьей строки наименьшее. Вычисление количества элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

8. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C . Вычислить квадрат той матрицы ($A * A$), в которой количество нулевых элементов побочной диагонали наибольшая. Вычисление количества элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

9. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C . Вычислить квадрат той матрицы ($A * A$), в которой количество нулевых элементов главной диагонали наибольшая. Вычисление количества элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

10. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C . Вычислить квадрат той матрицы ($A * A$), в которой произведение отрицательных элементов побочной диагонали наибольшая. Вычисление

количества элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

11. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C. Вычислить квадрат той матрицы (A*A), в которой среднее арифметическое отрицательных элементов побочной диагонали наибольшая. Вычисление количества элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

12. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C. Вычислить квадрат той матрицы (A*A), в которой произведение положительных элементов первого столбца наибольшая. Вычисление суммы элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

15. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C. Вычислить квадрат той матрицы (A*A), в которой среднее арифметическое положительных элементов первого столбца наибольшая. Вычисление суммы элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

13. **Задача 4 «Функции+ процедура».** Дано 3 квадратные матрицы A, B и C. Вычислить квадрат той матрицы (A*A), в которой произведение положительных элементов побочной диагонали наибольшая. Вычисление количества элементов оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры

14. **Задача 4 «Функции+ процедура»** Дано 3 квадратные матрицы A, B и C. Вычислить квадрат той матрицы (A*A), в которой след наименьший. Вычисление следа оформить в виде функции. Вычисление квадрата матрицы оформить в виде процедуры. (След матрицы – это сумма элементов главной диагонали)

ПЕРЕДАЧА ПАРАМЕТРОВ ПО ССЫЛКЕ

Рассмотрим третий случай *передачи параметров по ссылке (по адресу)* на примере этой же функции. Для передачи параметров по ссылке (*по адресу*) формальные параметры функции описываются как ссылки. (*как адреса переменных минуя указатели*)

Пример:

```
void change ( int &x, int &y){int temp = x ; x = y ; y = temp;}
void main()
{int a = 3, b = 5;...change( a, b );...}
```

При вызове функции формальному параметру-ссылке **x** присваивается значение переменной **a** (**x = a = 3**) и адрес переменной **a** (**&x = &a**) , что за собой повлечет изменение значения **a**, в главной функции.

ССЫЛКИ

Ссылка – модифицированная форма указателя и представляет собой *синоним* имени простой переменной.

Формат оператора объявления ссылки:

тип & имя;

Для того, чтобы инициализировать ссылку, ей нужно присвоить имя простой переменной и тогда она становится её синонимом.

Пример:

```
int a = 3; // описание целой переменной
```

```
int &p = a; /* описание ссылки p на целый тип и инициализация
ссылки переменной a */
```

После такого присвоения адрес ссылки **p** будет равен адресу переменной **a** и ее значение будет равно тоже значению переменной **a**. Таким образом, ссылка становится *псевдонимом* переменной **a**. Чаще всего ссылки используются в качестве *параметров функции*,

Пример. Описать функцию типа **void** (не возвращающую значение) для определения площади окружности по заданному радиусу. Использовать передачу параметра **S** по ссылке

```
void square_circle (float r, float &s)
{ s=2.0*3.14*r*r; }
int main()
{ float R1,S1;
  cout<<"vvedite R1=";
  cin>>R1;
  square_circle(R1,S1);
  cout<<"1-ay S1="<<S1;
  ..... float R2,S2;
  cout<<"vvedite R2=";
  cin>>R2;
  square_circle(R2,S2);
  cout<<"2-ay S2="<<S2;
}
```

Пример

```
float square_circle (float r)
{ float s=2.0*3.14*r*r;
  return(s); }
int main()
{ float R1,S1;
  cout<<"vvedite R1=";
  cin>>R1;
  S1=square_circle(R1);
  cout<<"1-ay S1="<<S1;
  ..... float R2,S2;
  cout<<"vvedite R2=";
```



```

cin>>R2;
S2= square_circle(R2);
cout<<"2-ay S2="<<S2;
}

```

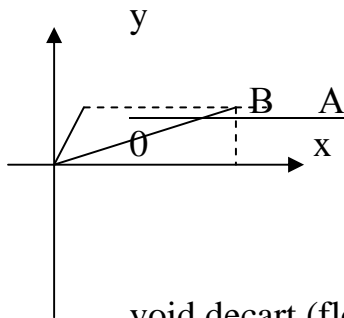
Пример Задача №1 Оформить процедуру переводящую полярные координаты точки в декартовые $(\rho, \varphi) \rightarrow (x, y)$ по формулам:

$$X = \rho * \cos \varphi$$

$$Y = \rho * \sin \varphi$$

В главной программе заданы полярные координаты двух точек на плоскости $A(\rho_1, \varphi_1)$, $B(\rho_2, \varphi_2)$.

Получить декартовы координат $A(x_1, y_1)$ $B(x_2, y_2)$ этих точек и выяснить, какая из них лежит ближе к оси ординат (то есть где меньше координата x).



```

void decart (float ro, float f, float* x, float *y)
{
*X = ro*cos(fi);
*Y = ro*sin(fi);
}
Int main()
{
float ro1, ro2, fi1, fi2, x1, y1, x2, y2; //глобальные переменные

cout<<"ввести полярные координаты двух точек";
cin>>ro1, fi1, ro2, fi2;
{1-ый вызов процедуры}
Decart(ro1, fi1, &x1, &y1);
{фактические параметры в той последовательности, в которой были
в описании процедуры}
{2-ый вызов процедуры}
Decart (ro2, fi2, &x2, &y2);
Cout<< "координаты первой точки= "<<x1, y1);
Cout<<"координаты второй точки= "<<x2, y2);
If (x1<x2)
Cout<< "первая точка ближе к оси ординат";
Else cout<<" вторая точка ближе к оси ординат";
}

```

}

ФУНКЦИИ СО СТРУКТУРАМИ.

Пример: Функция сортировка гор по возрастанию высот

```

#include<stdlib.h>
#include<time.h>
#include<conio.h>
#include<math.h>
#include<iostream.h>
#include<time.h>
#include<stdlib.h>
#include<string.h>
using namespace std;
struct T_mountain
{
    string name;
    int h;
};

void sort (T_mountain rok[],int n)
{
    for(int i=0;i<n-1;i++)
        for(int j=0;j<n-1;j++)
            if( rok[j].h>rok[j+1].h)
                {
                    T_mountain temp=rok[j];
                    rok[j]=rok[j+1];
                    rok[j+1]=temp;
                }
}

void vyvod (T_mountain rok[],int n)
{
    for(int i=0;i<n;i++)
        cout<<"имя="<<rok[i].name<<"  h= "<<rok[i].h<<endl;
}

int main()
{
    struct T_mountain rocks[10];
    int n, i;
    cout<<"vvedine n=";
    cin >>n;
    for (i = 0; i < n; i ++ )

```

```

{ cout<<"vvedite name";
  cin>>rocks[i].name ;
  cout<<"vvedite height";
  cin>>rocks[i].h ;
}

sort(rocks,n);
Cout<<endl;<<"список гор упорядоченный"<<endl;
vyvod(rocks,n);
getch();
}

```

Задания для лабораторной работы №4. Передача параметров по ссылке и по указателю

1.Задача 1 «Процедуры» Создать функцию типа void для поиска min и его номера в массиве. В главной программе Даны три вектора (массива) $x(n), y(n), z(n)$. Вызвать 3 раза функцию типа void. Далее Если наименьший элемент вектора x положителен и находится в о 2-ой половине этого вектора и если в векторе z наименьший элемент равен 8, тогда все элементы вектора y , предыдущие перед его наименьшим элементом, заменить на 1.

2.Задача 1 «Процедуры» Описать функцию типа void, которая присваивает параметру x квадрат меньшего числа из 2 $-x$ данных вещественных чисел x и y , а параметру y квадрат большего из чисел x и y . Применить функцию типа void для трех разных пар чисел a и b ; c и d ; p и q .

3.Задача 1 «Процедуры» Создать функцию типа void для поиска max и его номера в массиве . В главной программе Даны три вектора(массива) $x(n), y(n), z(n)$. Вызвать 3 раза функцию типа void. Далее Если наибольший элемент вектора x равен 7 и находится во 2-ой половине этого вектора и если в векторе y наибольший элемент равен 100, тогда все элементы вектора z , последующие за его наибольшим элементом, заменить на их модули.

4.Задача 1 «Процедуры» Дано два одномерных массива x и y одинаковой длины n . Описать функцию типа void, которая присваивает третьему массиву z сумму векторов (массивов) x и y . Применить функцию типа void для двух разных пар массивов a и b ; c и d .

5. Задача 1 «Процедуры» Создать функцию типа void для поиска min и его номера в массиве . В главной программе Даны три вектора (массива) $x(n), y(n), z(n)$. Вызвать 3 раза функцию типа void. Если наименьший элемент вектора x положителен и находится в о 2-ой половине этого вектора и если в векторе y наименьший элемент отрицателен, тогда все элементы вектора z , последующие за его наименьшим элементом, заменить число 100.

6.Задача 1 «Процедуры» Дано два одномерных массива x и y одинаковой длины n . Описать функцию типа void, которая присваивает

третьему массиву \mathbf{z} сумму квадратов векторов (массивов) \mathbf{x} и \mathbf{y} . Применить функцию типа `void` для двух разных пар массивов \mathbf{a} и \mathbf{b} ; \mathbf{c} и \mathbf{d} .

7.Задача 1 «Процедуры». Создать функцию типа `void` для поиска `max` и его номера в массиве. В главной программе Даны три вектора (массива) $x(n), y(n), z(n)$. Вызвать 3 раза функцию типа `void`. Если наибольший элемент вектора \mathbf{x} отрицателен и находится в первой половине этого вектора и если в векторе \mathbf{y} наибольший элемент равен 10, тогда все элементы вектора \mathbf{z} , предшествующие его наибольшему элементу, заменить на их квадраты,

8.Задача 1 «Процедуры»1. Дано два одномерных массива \mathbf{x} и \mathbf{y} одинаковой длины \mathbf{n} . Описать функцию типа `void`, которая присваивает третьему массиву \mathbf{z} разность векторов (массивов) \mathbf{x} и \mathbf{y} . Применить функцию типа `void` для двух разных пар массивов \mathbf{a} и \mathbf{b} ; \mathbf{c} и \mathbf{d} .

9.Задача 1 «Процедуры». Дано два одномерных массива \mathbf{x} и \mathbf{y} одинаковой длины \mathbf{n} . Описать функцию типа `void`, которая присваивает третьему массиву \mathbf{z} по элементное произведение векторов (массивов) \mathbf{x} и \mathbf{y} . Применить функцию типа `void` для двух разных пар массивов. \mathbf{a} и \mathbf{b} ; \mathbf{c} и \mathbf{d}

10. Задача 1 «Процедуры» Переменной t присвоить значение `true`, если уравнения $kx^2+3.5x+a^2=0$ и $2x^2+7x+k+2=0$ имеют вещественные корни и при этом оба корня первого уравнения больше чем оба корнями второго. Переменной t присвоить значение `false` во всех остальных случаях. Описать функцию типа `void` для решения квадратного уравнения вида $ax^2+bx+c=0$.

11. Задача 1 «Процедуры» Создать функцию типа `void` для поиска `min` и его номера в массиве. В главной программе Даны три вектора (массива) $x(n), y(n), z(n)$. Вызвать 3 раза функцию типа `void`. Если наименьший элемент вектора \mathbf{x} положителен и находится в первой половине этого вектора и если в векторе \mathbf{y} наименьший элемент больше 10, тогда все элементы вектора \mathbf{z} , последующие за его наименьшим элементом, заменить на их модули.

12. Задача 1 «Процедуры». Дано два одномерных массива \mathbf{x} и \mathbf{y} одинаковой длины \mathbf{n} . Описать функцию типа `void`, которая присваивает третьему массиву \mathbf{z} сумму векторов (массивов) \mathbf{x} и \mathbf{y} . Использовать созданную функцию типа `void` для вычисления $\mathbf{d}=\mathbf{a}+\mathbf{b}+\mathbf{c}$.

13. Задача 1 «Процедуры» Переменной t присвоить значение `true`, если уравнения $x^2+6.2x+a^2=0$ и $x^2+ax+b-1=0$ имеют вещественные корни и при этом оба корня первого уравнения лежат между корнями второго. Переменной t присвоить значение `false` во всех остальных случаях. Описать функцию типа `void` для решения квадратного уравнения вида $ax^2+bx+c=0$.

14. Задача 1 «Процедуры». Создать функцию типа `void` для поиска `max` и его номера в массиве. В главной программе Даны три вектора $x(n), y(n), z(n)$. Вызвать 3 раза функцию типа `void`. Если наибольший элемент вектора \mathbf{x} равен 10 и находится в первой половине этого вектора и если в векторе \mathbf{y} наибольший элемент отрицательный, тогда все элементы вектора \mathbf{z} , предшествующие его наибольшему элементу, заменить на их кубы, .

15. Задача 1 «Процедуры». Создать функцию типа `void` для поиска `min` и его номера в массиве. В главной программе Даны три вектора

(массива) $x(n), y(n), z(n)$. Вызвать 3 раза функцию типа void. Если наименьший элемент вектора x равен 5 и находится в первой половине этого вектора и если в векторе y наименьший элемент больше 10, тогда все элементы вектора z , последующие за его наименьшим элементом, заменить на их квадраты.

16. Задача 1 «Процедуры» Описать функцию типа void, которая присваивает параметру x большее число из 2 –х данных вещественных чисел x и y , а параметру y меньшее из чисел x и y . Применить функцию типа void для трех разных пар чисел a и b ; c и d ; p и q .

17. Задача 1 «Процедуры» Описать функцию типа void, которая присваивает параметру x большее число из 3 –х данных вещественных чисел x , y и z , параметру z меньшее из чисел x , y и z , а параметру y среднее из чисел x , y и z . Так что бы получилось $x > y > z$. Применить функцию типа void для двух разных троек чисел. a , b и c ; d , p и q .

Задача 2 "Процедуры сортировки массива"

1. Задача 2. Описать две функции типа void сортировки одномерного массива длиной N по убыванию и по возрастанию. Дано два массива a и b . Отсортировать a по возрастанию, b – по убыванию. Логической переменной t присвоить true, если наименьший элемент a больше наибольшего элемента b .

2. Задача 2. Описать функцию типа void сортировки одномерного массива длиной N по убыванию методом «Выбора и перестановки max». В главной программе Дано два массива a и b . Отсортировать a и, b – по убыванию. Поменять местами максимальный элемент массива a и второй наибольший (следующий за максимальным) в массиве b . Еще поменять местами второй наименьший массива a и наименьший массива b .

3. Задача 2. Описать функцию типа void, сортировки одномерного массива длиной N по убыванию методом «Пузырька». В главной программе Дано два массива a и b . Отсортировать a и, b – по убыванию. Определить, что больше наименьший элемент a или наибольшей элемент b .

4. Задача 2. Описать функцию типа void, сортировки одномерного массива длиной N по убыванию методом «Выбора и перестановки max». В главной программе Дано два массива a и b . Отсортировать a и, b – по убыванию. Вывести квадрат наибольшего элемента массива a , если второй наибольший (следующий за максимальным) в массиве b равен 20.

5. Задача 2. Описать функцию типа void, сортировки одномерного массива длиной N по возрастанию методом «Выбора и перестановки max». В главной программе Дано два массива a и b . Отсортировать a и, b – по возрастанию. Поменять местами второй наименьший массива a и наибольший массива b .

6. Задача 2. Описать функцию типа void, сортировки одномерного массива длиной N по убыванию методом «Пузырька». В главной программе Дано два массива a и b . Отсортировать a и, b – по убыванию. Если максимальный элемент массива b отрицателен, то заменить все элементы массива a на их модули кроме первого(max) и последнего (min)

7. **Задача 2.** Описать две процедуры, сортировки одномерного массива длиной N по убыванию и по возрастанию. Дано два массива x и y . Отсортировать x убыванию по, y – по возрастанию. Логической переменной t присвоить true, если наименьший элемент y больше наибольшего элемента x .

8. **Задача 2.** Описать функцию типа void, сортировки одномерного массива длиной N по убыванию методом «Простой перестановки». В главной программе Дано два массива a и b . Отсортировать a и, b – по убыванию. Если минимальный элемент массива a равен 1, то второй наибольший (следующий за максимальным) в массиве b заменить на 100.

9. **Задача 2.** Описать функцию типа void, сортировки одномерного массива длиной N по убыванию методом «Пузырька». В главной программе Дано два массива a и b . Отсортировать a и, b – по убыванию. Если минимальный элемент массива b отрицателен, то заменить все элементы массива a на их модули.

10. **Задача 2.** Описать функцию типа void, сортировки одномерного массива длиной N по убыванию методом «Выбора и перестановки .max».. В главной программе Дано два массива a и b . Отсортировать a и, b – по убыванию. Поменять местами минимальный элемент массива a и второй наибольший (следующий за максимальным) в массиве b .

11. **Задача 2.** Описать функцию типа void, сортировки одномерного массива длиной N по возрастанию методом «Пузырька». В главной программе Дано два массива a и b . Отсортировать a и, b – по возрастанию. Если максимальный элемент массива $a < 5$, то заменить все элементы массива b на их квадраты кроме первого (min) и последнего (max).

12. **Задача 2.** Описать функцию типа void, сортировки одномерного массива длиной N по возрастанию методом «Выбора и перестановки .max».. В главной программе Дано два массива a и b . Отсортировать a и, b – по возрастанию. Поменять местами второй наименьший массива a и наименьший массива b .

13. **Задача 2.** Описать две процедуры, сортировки одномерного массива длиной N по убыванию и по возрастанию. Дано два массива a и v . Отсортировать a по возрастанию, v – по убыванию. Логической переменной t присвоить true, если наименьший элемент a больше наибольшего элемента v .

14. **Задача 2.** Описать функцию типа void, сортировки одномерного массива длиной N по возрастанию методом «Простой перестановки». В главной программе Дано два массива a и b . Отсортировать a и, b – по возрастанию. Если минимальный элемент массива $a < 2$, то заменить все элементы массива b на их квадраты.

15. **Задача 2.** Описать две процедуры, сортировки одномерного массива длиной N по убыванию и по возрастанию. Дано два массива x и y . Отсортировать x убыванию по, y – по возрастанию. Логической переменной t присвоить true, если наименьший элемент y больше наибольшего элемента x .

Задача № 3 «Процедуры сортировки структурного массива»

1. Создать комбинированный (структурный) тип для расписания движения поездов (станция отправления, станция прибытия, время прибытия, время в пути). Описать процедуру сортировки комбинированного массива по названию станции отправления для одного массива. Пользователь задает два комбинированных массива по N элементов в каждом. (для электричек и поездов дальнего следования). Применить функцию два раза для заданных двух списков. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

2. Создать комбинированный (структурный) тип для списка книг домашней библиотеки (автор, название книги, издательство, год издания, стоимость). Описать процедуру сортировки одного списка по году издания. Пользователь задает два комбинированных массива по N элементов в каждом (для двух списков книг). Применить функцию два раза для заданных двух списков книг. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

3. Создать комбинированный (структурный) тип для расписания экзаменов и зачетов (предмет, вид отчетности, число, преподаватель). Описать процедуру сортировки по фамилии преподавателя в одном таком комбинированном массиве. Пользователь задает два комбинированных массива по N элементов в каждом.(для зимней и летней сессий). Применить функцию два раза для заданных двух сессий. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

4. Создать комбинированный (структурный) тип для сведения о студентах (Ф. И. О., курс, группа, номер зачетки, средний балл). Описать процедуру сортировки по алфавиту по фамилии в одном таком комбинированном массиве. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух университетов). Применить функцию два раза для заданных двух университетов. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

5. Создать комбинированный (структурный) тип для расписания учителя (номер урока, время начала урока, класс, предмет, номер кабинета). Описать процедуру сортировки по алфавиту по названию предмета в одном таком комбинированном массиве. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух учителей). Применить функцию два раза для заданных двух учителей. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

6. Создать комбинированный (структурный) тип для информации о наличии товаров на складе (наименование, артикул, дата получения, единица измерения, количество, цена). Описать процедуру сортировки по артикулу в

одном комбинированном массиве. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух складов). Применить функцию два раза для заданных двух складов. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

7. Создать комбинированный (структурный) тип для "Телефонный справочник" (Ф. И. О., адрес, номер телефона). Описать процедуру сортировки по фамилии в одном комбинированном массиве. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух пользователей). Применить функцию два раза для заданных двух пользователей. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

8. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать процедуру сортировки CD-дисков по стоимости дисков в коллекции. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

9. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать процедуру сортировки CD-дисков по имени исполнителя в коллекции. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

10. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать процедуру сортировки CD-дисков по названию альбома в коллекции. Пользователь задает два комбинированных массива по N элементов в каждом.(для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

11. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать процедуру сортировки дисков по году выпуска в коллекции. Пользователь задает два комбинированных массива по N элементов в каждом. (для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

12. Создать комбинированный (структурный) тип для списка CD-дисков (название альбома, исполнитель, стиль, год выпуска, длительность, стоимость). Описать процедуру сортировки дисков по длительности в

коллекции. Пользователь задает два комбинированных массива по N элементов в каждом. (для двух коллекций). Применить функцию два раза для заданных двух коллекций. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

13. Создать комбинированный (структурный) тип для списка книг домашней библиотеки (автор, название книги, издательство, год издания, стоимость). Описать процедуру сортировки одного списка по стоимости. Пользователь задает два комбинированных массива по N элементов в каждом (для двух списков книг). Применить функцию два раза для заданных двух списков книг. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

14. Создать комбинированный (структурный) тип для списка книг домашней библиотеки (автор, название книги, издательство, год издания, стоимость). Описать процедуру сортировки одного списка по автору. Пользователь задает два комбинированных массива по N элементов в каждом (для двух списков книг). Применить функцию два раза для заданных двух списков книг. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

15. Создать комбинированный (структурный) тип для списка книг домашней библиотеки (автор, название книги, издательство, год издания, стоимость). Описать процедуру сортировки одного списка по году издания. Пользователь задает два комбинированных массива по N элементов в каждом (для двух списков книг). Применить функцию два раза для заданных двух списков книг. . (Так же будет уместно описать процедуру ввода комбинированного массива и процедуру вывода.)

Задача №4 "Процедуры с матрицами"

1. **Задача 4.** Описать функцию типа void поиска минимального элемента и его номера в заданной строке матрицы.(номер строки – входной параметр) В главной программе Даны две матрицы $A(n \times n)$, $B(n \times n)$. Поменять местами минимальные элементы в первых строках данных матриц.

2. **Задача 4.** Описать функцию типа void, меняющую местами q -ую и s -ую строки матрицы $X(n \times n)$ (параметры q и s входные параметры процедуры) Применить функцию типа void для обмена 2-ой и последней строки в матрице A и для обмена 3-ей и 5-ой строки в матрице B .

3. **Задача 4.** Описать функцию типа void поиска минимального элемента и его 2-ух позиций во всей матрице. Даны две матрицы $A(n \times n)$, $B(n \times n)$. Поменять местами минимальные элементы матриц A и B .

4. **Задача 3** Даны три матрицы $A(n \times n)$, $B(n \times n)$, $C(n \times n)$ и три вектора $x(n)$, $y(n)$, $z(n)$. Вычислить $u = Ax + By - Cz + Bx$. Использовать функцию типа void для вычисления произведения матрицы на вектор

5. **Задача 3.** Описать функцию типа void поиска максимального элемента и его номера в заданном столбце матрицы .(номер столбца – входной параметр). В главной программе Даны две матрицы $A(n \times n)$, $B(n \times n)$.

Поменять местами максимальные элементы в первых столбцах данных матриц.

6. Задача 3. Описать функцию типа void отражения матрицы относительно главной диагонали. Даны две матрицы $A(n \times n)$, $B(n \times n)$., Определить можно ли отражением относительно главной диагонали преобразовать одну в другую.

7. Задача 3. Описать функцию типа void поиска максимального элемента и его номера в заданной строке матрицы. (номер строки – входной параметр) В главной программе Даны две матрицы $A(n \times m)$, $B(n \times m)$. Поменять местами максимальные элементы в 3-ой строке матрицы A и в 1-ей строке матрицы B.

8. Задача 3. Описать функцию типа void, меняющую местами первый и последний столбец матрицы $X(n \times n)$ Применить функцию типа void для двух разных матриц.

9. Задача 3. Заданы три матрицы $A(n)$, $B(n)$ и $C(n)$. Выяснить и напечатать, сколько из них являются симметрическими. (Матрица называется симметрической, если транспонированная матрица равна исходной). Транспонирование матрицы оформить в виде процедуры.

10. Задача 3 Известно ,что матрицах может быть только один отрицательный элемент. Описать функцию типа void поиска отрицательного элемента и его местонахождения в матрице. В главной программе Даны две матрицы $A(n \times n)$, $B(n \times n)$. Поменять местами отрицательные элементы этих матриц

11. Задача 3. Описать функцию типа void поиска первого положительного элемента и его номера в заданном столбце матрицы. (номер столбца – входной параметр) В главной программе Дана матрицы $A(n \times m)$. Поменять местами такие положительные элементы в первом и 3-ем столбцах матрицы.

12. Задача 3. Описать функцию типа void, меняющую местами k-ый и p-ый столбец матрицы $X(n \times n)$ (параметры k и p входные параметры) Применить функцию типа void для обмена 1-го и последнего столбца в матрице A и в матрице B.

13. Задача 3. Описать функцию типа void поиска последнего положительного элемента и его номера в заданной строке матрицы. (номер строки – входной параметр). В главной программе Даны две матрицы $A(n \times n)$, $B(n \times n)$. Поменять местами такие положительные элементы в первой строке матрицы A и в 3-ей строке матрицы B.

14. Задача 3. Даны три матрицы $A(n \times n)$, $B(n \times n)$, $C(n \times n)$ и три вектора $x(n)$, $y(n)$, $z(n)$. Вычислить $u = Ax + Bz - Cy$. Использовать функцию типа void для вычисления произведения матрицы на вектор.

15. Задача 3. Описать функцию типа void, меняющую местами p-ую и q-ую строки матрицы $X(n \times n)$ (параметры p и q входные параметры) Применить функцию типа void для обмена 1-ой и 3-ей строки в матрице A и для обмена 2-ой и последней строки в матрице B.

16. Задача 3. Описать функцию типа void поиска последнего

отрицательного элемента и его номера в заданном столбце матрицы. (номер столбца – входной параметр). В главной программе Даны две матрицы $A(n*m)$, $B(n*m)$. Поменять местами такие отрицательные элементы в 2-ом столбце матрицы A и в 4-ем столбце матрицы B .

17. Задача 3 . Описать функцию типа `void` поиска максимального элемента и его 2-ух позиций во всей матрице. Даны две матрицы $A(n \times n)$, $B(n \times n)$. Поменять местами максимальные элементы матриц A и B .

18. Задача 3 . Даны матрицы $A(n \times n)$, $B(n \times n)$ и два вектора $x(n), y(n)$. Вычислить $u = Ay + Bx$. Использовать функцию типа `void` для вычисления произведения матрицы на вектор.

Задача №5 "Процедуры"

17. . Описать **функцию типа `void`** `MinmaxNum(A, N, NMin, NMax)`, находящую номера минимального и максимального элементов массива A размера N . Выходные параметры целого типа: `NMin` (номер минимального элемента) и `NMax` (номер максимального элемента). С помощью этой процедуры найти номера минимальных и максимальных элементов массивов A, B, C

18. Описать **функцию типа `void`** `Invert(A, N)`, меняющую порядок следования элементов массива A размера N на противоположный (инвертирование массива). Массив A является входным и выходным параметром. С помощью этой процедуры инвертировать массивы A, B, C .

19. . Описать **функцию типа `void`** `Smooth1(A, N)`, выполняющую сглаживание массива A размера N следующим образом: элемент A_k заменяется на среднее арифметическое первых k исходных элементов массива A . Массив A является входным и выходным параметром. С помощью этой процедуры выполнить пятикратное сглаживание данного массива A размера N , выводя результаты каждого сглаживания.

20. Описать **функцию типа `void`** `Smooth2(A, N)`, выполняющую сглаживание массива A размера N следующим образом: элемент A_1 не изменяется, элемент A_k ($k = 2, \dots, N$) заменяется на полусумму исходных элементов A_{k-1} и A_k . Массив A является входным и выходным параметром. С помощью этой процедуры выполнить пятикратное сглаживание данного массива A размера N , выводя результаты каждого сглаживания.

21. Описать **функцию типа `void`** `Smooth3(A, N)`, выполняющую сглаживание массива A размера N следующим образом: каждый элемент массива заменяется на его среднее арифметическое с соседними элементами (при вычислении среднего арифметического используются исходные значения соседних элементов). Массив A является входным и выходным параметром. С помощью этой процедуры выполнить пятикратное

сглаживание данного массива A размера N , выводя результаты каждого сглаживания.

22. Описать **функцию типа void** `RemoveX(A, N, X)`, удаляющую из целочисленного массива A размера N элементы, равные целому числу X . Массив A и число N являются входными и выходными параметрами. С помощью этой процедуры удалить числа X_A , X_B , X_C из массивов A , B , C размера N_A , N_B , N_C соответственно и вывести размер и содержимое полученных массивов.

23. Описать **функцию типа void** `RemoveForInc(A, N)`, удаляющую из вещественного массива A размера N «лишние» элементы так, чтобы оставшиеся элементы оказались упорядоченными по возрастанию: первый элемент не удаляется, второй элемент удаляется, если он меньше первого, третий — если он меньше предыдущего элемента, оставленного в массиве, и т. д. Например, массив 5.5, 2.5, 4.6, 7.2, 5.8, 9.4 должен быть преобразован к виду 5.5, 7.2, 9.4. Массив A и число N являются входными и выходными параметрами. С помощью этой процедуры преобразовать массивы A, B, C размера N_A, N_B, N_C соответственно и вывести размер и содержимое полученных массивов.

24. Описать **функцию типа void** `DoubleX(A, N, X)`, дублирующую в целочисленном массиве A размера N элементы, равные целому числу X . Массив A и число N являются входными и выходными параметрами. С помощью этой процедуры продублировать числа X_A , X_B , X_C в массивах A, B, C размера N_A, N_B, N_C соответственно и вывести размер и содержимое полученных массивов.

25. Описать **функцию типа void** `SortArray(A, N)`, выполняющую сортировку по возрастанию вещественного массива A размера N . Массив A является входным и выходным параметром. С помощью этой процедуры отсортировать массивы A, B, C размера N_A, N_B, N_C соответственно.

26. Описать **функцию типа void** `SortIndex(A, N, I)`, формирующую для вещественного массива A размера N индексный массив I — массив целых чисел того же размера, содержащий номера элементов массива A в том порядке, который соответствует возрастанию элементов массива A (сам массив A при этом не изменяется). Индексный массив I является выходным параметром. С помощью этой процедуры создать индексные массивы для массивов A, B, C размера N_A, N_B, N_C соответственно.

27. Описать **функцию типа void** `Bell(A, N)`, меняющую порядок элементов вещественного массива A размера N на следующий: наименьший элемент массива располагается на первом месте, наименьший из оставшихся элементов — на последнем, следующий по величине располагается на

втором месте, следующий — на предпоследнем и т. д. (в результате график значений элементов будет напоминать колокол). Массив A является входным и выходным параметром. С помощью этой процедуры преобразовать массивы A , B , C размера NA , NB , NC соответственно.

28. Описать **функцию типа void** `Split1(A, NA, B, NB, C, NC)`, формирующую по вещественному массиву A размера NA два вещественных массива B и C размера NB и NC соответственно; при этом массив B содержит все элементы массива A с нечетными порядковыми номерами (1, 3, : : :), а массив C — все элементы массива A с четными номерами (2, 4, : : :). Массивы B и C и числа NB и NC являются выходными параметрами. Применить эту функцию типа `void` к данному массиву A размера NA и вывести размер и содержимое полученных массивов B и C .

29. . Описать **функцию типа void** `Split2(A, NA, B, NB, C, NC)`, формирующую по целочисленному массиву A размера NA два целочисленных массива B и C размера NB и NC соответственно; при этом массив B содержит все четные числа из массива A , а массив C — все нечетные числа (в том же порядке). Массивы B и C и числа NB и NC являются выходными параметрами. Применить эту функцию типа `void` к данному массиву A размера NA и вывести размер и содержимое полученных массивов B и C .

30. Описать **функцию типа void** `ArrayToMatrRow(A, K, M, N, B)`, формирующую по вещественному массиву A размера K матрицу B размера $M \times N$ (матрица заполняется элементами массива A по строкам). «Лишние» элементы массива игнорируются; если элементов массива недостаточно, то оставшиеся элементы матрицы полагаются равными 0. Двумерный массив B является выходным параметром. С помощью этой процедуры на основе данного массива A размера K и целых чисел M и N сформировать матрицу B размера $M \times N$.

31. Описать **функцию типа void** `ArrayToMatrCol(A, K, M, N, B)`, формирующую по вещественному массиву A размера K матрицу B размера $M \times N$ (матрица заполняется элементами массива A по столбцам). «Лишние» элементы массива игнорируются; если элементов массива недостаточно, то оставшиеся элементы матрицы полагаются равными 0. Двумерный массив B является выходным параметром. С помощью этой процедуры на основе данного массива A размера K и целых чисел M и N сформировать матрицу B размера $M \times N$.

32. Описать **функцию типа void** `Chessboard(M, N, A)`, формирующую по целым положительным числам M и N матрицу A размера $M \times N$, которая содержит числа 0 и 1, расположенные в «шахматном» порядке, причем $A_{1;1} = 0$. Двумерный целочисленный массив A является

выходным параметром. С помощью этой процедуры по данным целым числам M и N сформировать матрицу A размера $M \times N$.

Дополнительная * Задача №1*.

1. Описать функцию типа `void DigitCountSum(K, C, S)`, находящую количество C цифр целого положительного числа K , а также их сумму s (K — входной, C и S — выходные параметры целого типа). С помощью этой процедуры найти количество и сумму цифр для каждого из пяти данных целых чисел.

2. Описать функцию типа `void InvertDigits(K)`, меняющую порядок следования цифр целого положительного числа K на обратный (K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры поменять порядок следования цифр на обратный для каждого из пяти данных целых чисел.

3. Описать функцию типа `void AddRightDigit(D, K)`, добавляющую к целому положительному числу K справа цифру D (D — входной параметр целого типа, лежащий в диапазоне 0-9, K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры последовательно добавить к данному числу K справа данные цифры $D1$ и $D2$ выводя результат каждого добавления.

4. Дана квадратная матрица A . Написать программу вычисления матрицы $B = A^T * A^T * A * A$, где A^T - транспонированная матрица. Вычисление произведения матриц оформить в виде процедуры.

5. Описать функцию типа `void AddLeftDigit(D, K)`, добавляющую к целому положительному числу K слева цифру D (D — входной параметр целого типа, лежащий в диапазоне 1-9, K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры последовательно добавить к данному числу K слева данные цифры $D1$ и $D2$ выводя результат каждого добавления.

6. Даны две матрицы A и B . Написать программу, меняющую местами максимальные элементы этих матриц. Даны две матрицы размером 3×3 . Написать программу получения коммутатора $AB - BA$ и антикоммутатора $AB + BA$ этих матриц. Вычисление произведения матриц оформить в виде процедуры

7. Описать функцию типа `void DigitCountSum(K, C, S)`, находящую количество C цифр целого положительного числа K , а также их сумму s (K — входной, C и S — выходные параметры целого типа). С помощью этой процедуры найти количество и сумму цифр для каждого из пяти данных целых чисел.

8. Описать функцию типа `void InvertDigits(K)`, меняющую порядок следования цифр целого положительного числа K на обратный (K —

параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры поменять порядок следования цифр на обратный для каждого из пяти данных целых чисел.

9. Описать функцию типа `void SORTINC(A, B, C)`, меняющую содержимое переменных A, B, C таким образом, чтобы их значения оказались упорядоченными по возрастанию (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой процедуры упорядочить по возрастанию два данных набора из трех чисел: (A_1, B_1, C_1) и (A_2, B_2, C_2) .

10. Описать функцию типа `void SORTDEC3(A, B, C)`, меняющую содержимое переменных A, B, C таким образом, чтобы их значения оказались упорядоченными по убыванию (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой процедуры упорядочить по убыванию два данных набора из трех чисел: (A_1, B_1, C_1) и (A_2, B_2, C_2) .

11. Описать функцию типа `void DigitCountSum(K, C, S)`, находящую количество C цифр целого положительного числа K , а также их сумму s (K — входной, C и S — выходные параметры целого типа). С помощью этой процедуры найти количество и сумму цифр для каждого из пяти данных целых чисел.

12. Описать функцию типа `void InvertDigits(K)`, меняющую порядок следования цифр целого положительного числа K на обратный (K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры поменять порядок следования цифр на обратный для каждого из пяти данных целых чисел.

13. Описать функцию типа `void AddRightDigit(D, K)`, добавляющую к целому положительному числу K справа цифру D (D — входной параметр целого типа, лежащий в диапазоне 0-9, K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры последовательно добавить к данному числу K справа данные цифры D_1 и D_2 выводя результат каждого добавления.

14. Описать функцию типа `void DigitCountSum(K, C, S)`, находящую количество C цифр целого положительного числа K , а также их сумму s (K — входной, C и S — выходные параметры целого типа). С помощью этой процедуры найти количество и сумму цифр для каждого из пяти данных целых чисел.

ПОНЯТИЕ РЕКУРСИИ..

Функции и процедуры производящие вызов «самих себя» называются рекурсивными. *Рекурсивными называются функции, содержащие в себе операторы обращения к самим же себе.* Функции такого рода используются для создания динамической памяти, для разработки компилятора (СПО) и

т.д. Пока будем рассматривать тривиальные задачи, решаемые с помощью рекурсивных функций.

Пример 1 .

Вычисление факториала можно рассматривать не только как прямое перемножение всех целых чисел $n! = 1*2*3*3*...*n$, но и с обратной стороны, Можно свести задачу вычисления факториала , через предыдущий факториал $n! = (n-1)!*n$. (*чайник поставить и вскипятить*) Другими словами вычисление факториала можно свести к произведению двух величин, одна из которых определяется *через него же самого* (т.е. через факториал). Такое вычисление называется *рекурсивным*.

```
int Factorial(int n)
{
  if (n ==1) return(1);
  else
  return( Factorial(n-1) *n);
}
Int main()
{
  x= factorial(4); cout<<"x="<<x;
  int k; cin >>k;
  y= factorial(k);cout<<"y="<<y;
}
```

Чтобы понять как она работает, нужно учесть , что на время выполнения функции вызывающая программа приостанавливается, а в оперативной памяти ПК выделяется место для параметров вызванной функции (*и локальных переменных*). Таким образом , при вызовах функций накапливаются приостановленные программы, а также копии параметров функции (*и локальных переменных*)

Рекурсивная функция осуществляет многократный переход от текущего уровня алгоритма к более низкому уровню. Последовательно до тех пор пока не будет получено тривиальное решение ($0!=1$)

Рекурсивная функция – это своеобразный цикл от n до 1 с шагом -1.

Трассировка:

Y:= fact(4) = fact (4-1)=

↓ Fact(3) = fact(3-1)*3

↓ Fact(2) = fact(2-1)*2

↓ Fact(1)=1.

↓ Прямой ход рекурсии

Y:=24

Fact(4) = fact(3)*4 = 6*4= 24

Fact(3)= fact(2)*3= 2*3=6

Fact(2)= fact(1)*2 = 1*2=2.

Fact(1)= 1.

↑ Обратный ход рекурсии

Во время прямого хода функция углубляется сама в себя. При этом никаких вычислений не происходит, а происходят только вызовы функции. Во время обратного хода, когда n становится равной 1, получаем тривиальное решение $1!=1$ и происходит восхождение вверх, во время которого происходит вычисление. В нашем варианте – умножение на 2, на 3, на 4.

РЕКУРСИВНЫЕ ФУНКЦИИ

Пример2. Создать функцию для вычисления n -ного члена арифметической прогрессии.

$$a_n = a_{n-1} + d \quad a_0 = 2 \quad d=5$$

$$a = 2; 7; 12; 17; 22; \dots$$

```
Int arifm(int n)
{Int d=5;
  If( n== 0) return(2); Else return(arifm(n-1) +d);
}
```

```
Int main()
{int X=arifm(5); cout<<"x= "<<x;
Int n; cin>>n; int Y= arifm(n); cout<<"y="<<y;
}
```

Трассировка

X=arifm(5)
 Arifm=arifm(5-1)+5 = arifm(4)+5 прямой ход рекурсии
 Arifm(4)= arifm(4-1)+5 = arifm(3)+5
 Arifm(3)= arifm(3-1)+5 = arifm(2)+5
 Arifm(2)= arifm(2-1)+5=
 Arifm(1)= arifm(1-1)+5= ↓
 Arifm(0)=2.

Пример 3. Описать рекурсивную функцию для вычисления n -го члена ряда

40, 20, 10, 5, 5/2, 5/4, ... 5/128

Решение:

$$40/2=20$$

$$20/2=10$$

$$10/2=5$$

$$5/2$$

$$(5/2)/2=5/4$$

.....

Геометрическая прогрессия $B_1=40$; $d=1/2$;

$b_n=b_{n-1} * d$; - рекуррентная формула

```

float geom(int n)
{   If (n == 1) return( 40);   Else return(geom(n-1)/2.0);
}
Int main()
{
Float X=geom(4);
cout<<"x= "<<x);//5
Readln (n);
Y:= a(n);
Writeln ('y= ',y);
End.

```

Пример 4

$$s = \sum_{n=1}^{20} 2\sqrt{n}$$

Решение :

$$S_0=0$$

$$S_1=0+2$$

$$S_2=0+2+2\sqrt{2} = s_1+2\sqrt{2}$$

$$S_3=0+2+2\sqrt{2}+2\sqrt{3} = s_2+2\sqrt{3}$$

.....

$s_n = s_{n-1} + 2\sqrt{n}$ – рекуррентная формула

if(n==0) return(0); else return(sum(n-1)+2*sqrt(n));

Пример 5

$$S=2^2+4^2+6^2+8^2+\dots+10^2\dots$$

Решение:

$$S_0=0$$

$$S_1=0+2^2 = s_0+2^2$$

$$S_2=0+2^2+4^2 = s_1+4^2$$

$$S_3=0+2^2+4^2+6^2 = s_2+6^2$$

$s_n = s_{n-2} + n^2$ – рекуррентная формула

if(n==0) return(0); else return(sum(n-2)+pow(n,2));

Пример 6

$$p = \prod_{n=3}^{10} \frac{1}{n^2 + 1} \quad \text{if}(n==2) \text{ return}(1); \text{ else return}(pr(n-1)*1.0/(n*n+1));$$

Пример 7. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции: x – вещественное и k – целое

$$B) S = \frac{\sin x}{2} + \frac{\sin 2x}{3} + \frac{\sin 3x}{4} + \dots + \frac{\sin 9x}{10}$$

Решение:

$$S_n = S_{n-1} + \sin(n*x)/(n+1) \quad - \quad \text{рекуррентная формула}$$

Float Sum(float x, int k)

```
{
  If (k == 0) return( 0); Else return (sum(x, k -
1)+sin(k*x)*1.0/(k+1.0));
}
Begin
read(x);
Q:=s(x,3);
Writeln ('Q= ',Q);
End.
```

Пример 8

числа Фибоначчи

$$f_n = f_{n-1} + f_{n-2}$$

$$f_0 = 1, f_1 = 1;$$

далее *Решение:*

$$f_2 = 1 + 1 = 2$$

$$f_3 = 1 + 2 = 3$$

$$f_4 = 2 + 3 = 5$$

$$f_5 = 3 + 5 = 8$$

$$f_6 = 5 + 8 = 13$$

If ((n==1)|| (n==0)) return(1); else return(f(n-1)+f(n-2));

Пример 9.

$$x_i = \frac{x_{i-1}}{2} + \frac{2}{x_{i-2}}$$

$$X_1 = 1, X_2 = 1$$

далее

$$X_3 = 1/2 + 2/1 = 2,5$$

$$X_4 = 1/2 + 2/2,5$$

.....
 If ((n==1)||n==0) return(1); else return(f(n-1)/2+2/f(n-2));

Пример 10

$T_n = 3 * T_{n-1} + 2 * T_{n-2}$; при $T_0=1$; $T_1=2$;

If (n==0) return (1); else {
 if(n==1) return(2); else return(f(n-1)/2+2/f(n-2));}

Пример 11

$$Y = \sqrt{100 + \sqrt{99 + \dots + \sqrt{2 + \sqrt{1}}}}$$

Решение:

$$Y = \sqrt{1}$$

$$Y_2 = \sqrt{2 + \sqrt{1}}$$

$$Y_3 = \sqrt{3 + \sqrt{2 + \sqrt{1}}} = \sqrt{3 + Y_2}$$

$$Y_4 = \sqrt{4 + \sqrt{3 + \sqrt{2 + \sqrt{1}}}} = \sqrt{4 + Y_3}$$

$Y_n = \sqrt{n + Y_{n-1}}$ - рекуррентная формула

if(n==1) return(1); else return(sqrt(n+f(n-1)));}

Задания для лабораторной работы №5. Рекурсивные функции

ЗАДАЧА №1

1. Описать рекурсивную функцию для вычисления n –го члена ряда **3, 0.3, 0.03, 0.003, ... 0.00003**
2. Описать рекурсивную функцию для вычисления n –го члена ряда **10, 5, 5/2, 5/4, 5/8, 5/16, ...5/256**
3. Описать рекурсивную функцию для вычисления n –го члена ряда **6, 3, 3/2, 3/4, 3/8, ... 3/256**
4. Описать рекурсивную функцию для вычисления n –го члена ряда **1, 2, 4, 8,256**

5. Описать рекурсивную функцию для вычисления n -го члена ряда **2, 4, 6, ..., 20**
6. Описать рекурсивную функцию для вычисления n -го члена ряда **1, 3, 5, ..., 21**
7. Описать рекурсивную функцию для вычисления n -го члена ряда **1, 1.2, 1.4, 1.6, ... 3.0**
8. Описать рекурсивную функцию для вычисления n -го члена ряда **0.7 ; 0.07 ; 0.007 ; 0.0007; ...0.00000000007**
9. Описать рекурсивную функцию для вычисления n -го члена ряда **1, 3, 9, 27, 81729**
10. Описать рекурсивную функцию для вычисления n -го члена ряда **7, 17, 27, 37, 97**
11. Описать рекурсивную функцию для вычисления n -го члена ряда **3, 3.3, 3.6, 3.9, 4.2,6**
12. Описать рекурсивную функцию для вычисления n -го члена ряда **20, 10, 5, 5/2, 5/4, 5/8, 5/16, ...5/128**
13. Описать рекурсивную функцию для вычисления n -го члена ряда **5,2; 5,4; 5,6; 5,8; 6; 6,2; ... 10**
14. Описать рекурсивную функцию для вычисления n -го члена ряда **40, 20, 10, 5, 5/2, 5/4, ... 5/128**
15. Описать рекурсивную функцию для вычисления n -го члена ряда **1, 4, 16, 64, 256, 1024, 4096**

ЗАДАЧА №2

1. .Описать рекурсивную функцию для вычисления произведения
Вычислить : $y = \prod_{i=1}^{50} (i+0.5)$
2. Описать рекурсивную функцию для вычисления произведения
Вычислить : $y = \prod_{x=3}^{30} \sqrt{x+1}$

3. Описать рекурсивную функцию для вычисления произведения
Вычислить : $y = \prod_{x=2}^{15} (x^2 + \log_2 x)$
4. Описать 2 рекурсивные функции для вычисления произведения и суммы . В главной программе вызвать эти функции для вычисления произведения : $y = m \prod_{k=3}^8 \frac{1}{k} + \frac{1}{m} * \sum_{p=1}^5 \frac{a^p}{\sin p}$
5. Описать 2 рекурсивные функции для вычисления произведения и суммы . В главной программе вызвать эти функции для вычисления выражения : $y = \sqrt[3]{\prod_{x=3}^9 \frac{2 \lg x}{x}} * \frac{35,29}{\sum_{j=1}^7 j}$
6. Описать 2 рекурсивные функции для вычисления произведения и суммы . В главной программе вызвать эти функции для вычисления выражения : $y = \sqrt[3]{\prod_{k=1}^4 a^k * \sum_{j=6}^9 a^{-j}}$
7. Описать 2 рекурсивные функции для вычисления произведения и суммы . В главной программе вызвать эти функции для вычисления выражения : $y = \prod_{k=1}^8 (k + 2^k) + \sum_{j=3}^9 \sin 2j$
8. Описать рекурсивную функцию для вычисления суммы
Вычислить : $y = \sum_{x=2}^{10} e^{x + \log_2 x}$
9. Описать рекурсивную функцию для вычисления произведения
Вычислить : $y = \prod_{x=2}^{15} \frac{\ln x}{5}$
10. Описать рекурсивную функцию для вычисления произведения
Вычислить : $y = \prod_{x=3}^{30} 2\sqrt{x}$
11. Описать рекурсивную функцию для вычисления суммы
Вычислить : $y = \sum_{x=2}^{12} \log_5 x$

12. Описать рекурсивную функцию для вычисления произведения
 Вычислить : $y = \prod_{x=3}^{25} 5 \sin 2x$

13. Описать рекурсивную функцию для вычисления суммы
 Вычислить : $y = \sum_{x=1}^{15} \cos 5x * \sin 2x$

14. Описать рекурсивную функцию для вычисления произведения
 Вычислить : $y = \prod_{n=1}^{12} n^x$

ЗАДАЧА №3

1. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции : x – вещественное и k – целое
 $S = x - \frac{x}{4} + \frac{x}{9} - \frac{x}{16} + \frac{x}{25} - \dots + \frac{x}{100}$

2. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции: x – вещественное и k – целое
 $S = \sin x + \sin 2x + \sin 3x + \dots + \sin 7x$

3. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции: x – вещественное и k – целое
 $S = \cos x + 2 \cos 2x + 3 \cos 3x + \dots + 6 \cos 6x$

4. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции: x – вещественное и k – целое
 $S = \frac{e^{-x}}{2} + \frac{e^{-2x}}{3} + \frac{e^{-3x}}{4} + \dots + \frac{e^{-8x}}{9}$

5. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры : функции x – вещественное и k – целое
 $S = \sin 3x + \sin 4x + \sin 5x + \dots + \sin 11x$

6. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции: x – вещественное и k – целое
 $S = \cos 2x - \cos 3x + \cos 4x + \dots + \cos 10x$

7. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции: x – вещественное и k – целое
 $S = 2x + \frac{3x}{2} + \frac{4x}{3} + \frac{5x}{4} + \dots + \frac{11x}{10}$

8. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции: x – вещественное и k –

$$S = \frac{\sin x}{2} + \frac{\sin 2x}{3} + \frac{\sin 3x}{4} + \dots + \frac{\sin 9x}{10}$$

целое

9. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции: x – вещественное и k – целое

$$S = \cos x + \cos 2x + \cos 3x + \dots + \cos 13x$$

10. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции: x – вещественное и k –

$$S = \sin x + \frac{\sin 2x}{4} + \frac{\sin 3x}{9} + \frac{\sin 4x}{16} + \dots + \frac{\sin 9x}{81}$$

целое

11. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции: x – вещественное и k –

$$S = \frac{\cos x}{2} + \frac{\cos 2x}{4} + \frac{\cos 3x}{6} + \dots + \frac{\cos 8x}{16}$$

целое

12. Описать рекурсивную функцию для вычисления суммы первых k слагаемых. Входные параметры функции: x – вещественное и k –

$$\text{целое } S = 2 \cos x + 3 \cos 2x + 4 \cos 3x + \dots + 8 \cos 7x$$

13. Описать рекурсивную функцию для вычисления произведения первых n сомножителей. $p = (1 + 2/2^3)(1 + 2/3^3)(1 + 2/4^3)\dots(1 + 2/n^3)$

14. Описать рекурсивную функцию для вычисления произведения n нечетных натуральных чисел $p = (2n - 1)!! = 1 * 3 * 5 * 7 * \dots * (2n - 1)$

15. Описать рекурсивную функцию для вычисления произведения n четных натуральных чисел $p = (2n)!! = 2 * 4 * 6 * 8 * \dots * (2n)$

ЗАДАЧА №4

1. Описать рекурсивную процедуру для вычисления i –го члена ряда по формуле: $x_i = \frac{x_{i-1}}{2} + \frac{a}{2 \cdot x_{i-1}}$, $x_1 = 1$, $a > 0$

2. Описать рекурсивную функцию для вычисления n –го члена ряда по формуле: $T_n = 3 \cdot T_{n-1} + 2 \cdot T_{n-2}$; $T_0 = 1$; $T_1 = 2$;

3. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $T_n = 3 \cdot T_{n-1} \cdot T_{n-2}^5$; $T_0 = 2$; $T_1 = 2$;

4. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $R_n = 2 \cdot R_{n-1} \cdot 5 \cdot R_{n-2} + R_{n-1}^2$; $R_0 = 1$; $R_1 = 1$;

5. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $R_n = (2 \cdot R_{n-1} + R_{n-2})^3$; $R_0 = 2$; $R_1 = 1$;

6. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $R_n = (2 \cdot R_{n-1} + 4 \cdot R_{n-2}) \cdot 10 + R_{n-1}$; $R_0 = 5$; $R_1 = 3$;

7. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $k_n = 12 \cdot k_{n-1} + 6 \cdot k_{n-2}$; $k_0 = 1$; $k_1 = 5$;

8. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $z_n = 2 \cdot z_{n-1} + 6 \cdot z_{n-2}^2$; $z_0 = 5$; $z_1 = 1$

9. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $y_n = 7 \cdot y_{n-1} + 3/y_{n-2} + 15$; $y_0 = 1$; $y_1 = 2$;

10. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $y_n = 3 \cdot (y_{n-2} + 3 \cdot y_{n-1}^3) + 6$; $y_0 = 0$; $y_1 = 1$;

11. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $z_n = a \cdot z_{n-2} + b \cdot z_{n-1}^5$; $z_0 = 5$; $z_1 = 1$

12. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $z_n = a \cdot z_{n-1}^2 + b \cdot z_{n-2}$; $z_0 = 4$; $z_1 = 0$

13. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $y_n = 7 \cdot \sqrt{y_{n-1} + 3/y_{n-2} + 15}$; $y_0 = 1$; $y_1 = 2$;

14. Описать рекурсивную функцию для вычисления n -го члена ряда по формуле: $y_n = 3 \cdot (\sqrt{y_{n-2} + 3 \cdot y_{n-1}^3}) + 6$; $y_0 = 0$; $y_1 = 1$;

ЗАДАЧА №1*

1 Описать рекурсивную функцию для вычисления $y = \cos(15 + \cos(14 + \cos(\dots + \cos(2 + \cos(1))\dots))$

2 Описать рекурсивную функцию для вычисления $y = \sqrt{1 + \sqrt{2 + \dots + \sqrt{14 + \sqrt{15}}}}$

3 Описать рекурсивную функцию для вычисления
 $y = \sqrt{x^5 + \sqrt{x^4 + \dots + \sqrt{x+1}}}$ (не использовать: $x^n = \text{pow}(x, n)$)

4 Описать рекурсивную функцию для вычисления
 $y = ((\dots((\sin x^{25} + x)\sin x^{24} + x)\sin x^{23} + \dots + x)\sin^2 + x)\sin x + x)$ X - произвольное
число.

5 Описать рекурсивную функцию для вычисления
 $Y = \sqrt{3 + \sqrt{6 + \sqrt{9 + \dots + \sqrt{96 + \sqrt{99}}}}}$

6 Описать рекурсивную функцию для вычисления
 $Y = \sqrt{99 + \sqrt{96 + \dots + \sqrt{6 + \sqrt{3}}}}$

7 Описать рекурсивную функцию для вычисления
 $Y = \sqrt{2 + \sqrt{4 + \sqrt{8 + \dots + \sqrt{512 + \sqrt{1024}}}}}$

8 Описать рекурсивную функцию для вычисления
 $Y = \sqrt{1024 + \sqrt{512 + \dots + \sqrt{4 + \sqrt{2}}}}$

9 Описать рекурсивную функцию для вычисления
 $Y = \sqrt{\sin X + \sqrt{\sin 2X + \dots + \sqrt{\sin 98X + \sqrt{\sin 99x}}}}$

10 Описать рекурсивную функцию для вычисления
 $Y = \sqrt{\sin 99x + \sqrt{\sin 98X + \dots + \sqrt{\sin 2X + \sqrt{\sin X}}}}$

11 Описать рекурсивную функцию для вычисления
 $Y = \sqrt{\sin(X + \sqrt{\sin(2X + \dots + \sqrt{\sin(98X + \sqrt{\sin 99X}) \dots})})}$

12 Описать рекурсивную функцию для вычисления
 $Y = \sqrt{\sin(X + \sqrt{\sin 2(X + \dots + \sqrt{\sin 98(X + \sqrt{\sin 99X}) \dots})})}$

13 Описать рекурсивную функцию для вычисления
 $Y = \sin \sqrt{X + \sin \sqrt{2X + \dots + \sin \sqrt{98X + \sin \sqrt{99X}}}}$

14 Описать рекурсивную функцию для вычисления
 $Y = \sin \sqrt{X + \sin 2 \sqrt{X + \dots + \sin 98 \sqrt{X + \sin 99 \sqrt{X}}}}$

ЗАДАЧА *№2*
рекурсивные функции **.

Вариант 1. Напишите рекурсивную функцию для нахождения биномиальных коэффициентов (для заданного $M \geq i \geq j > 0$ вычислите все C_i^j).

$$C_n^m = \begin{cases} 1, & \text{при } m = 0, n > 0 \text{ или } n = m \geq 0 \\ 0, & \text{при } m > n > 0 \\ C_{m-1}^{n-1} + C_m^{n-1}, & \text{иначе} \end{cases}.$$

Вариант 2. Написать рекурсивную функцию вычисления *Наибольшего общего делителя* по алгоритму Эйлера:

Если B делится на A нацело, то $\text{НОД}(A, B) = A$.

В противном случае $\text{НОД}(A, B) = \text{НОД}(B \bmod A, A)$.

Вариант 3. Напишите две функции вычисления i -го числа Фибоначчи: рекурсивную и не рекурсивную. Напечатайте таблицу для сравнения времени вычисления i -го числа. Вычислите $f(k)$, $k = 15, 20, 30, 40$. Функция $f(n)$ определена для целых чисел следующим образом:

$$f(n) = \begin{cases} 1, & n = 1 \\ \sum_2^n f(n \operatorname{div} 2), & n \geq 2 \end{cases}.$$

Вариант 4. Создайте программу вычисления числа сочетаний из N по M . Число сочетаний определяется по формуле $C_N^M = \frac{N!}{M!(N-M)!}$, где N - количество элементов перебора. Для вычисления факториала напишите рекурсивную функцию.

Вариант 5. Для заданных m и x вычислить бином Ньютона $(1+x)^m$ непосредственно и по формуле разложения в ряд: $(1+x)^m = \sum_{i=0}^m C_m^i \cdot x^i$. Для вычисления C_m^i использовать рекуррентное соотношение (написать рекурсивную функцию): $C_m^{i+1} = C_m^i \frac{m-i}{i+1}$; $C_m^0 = 1$.

Вариант 6. Написать рекурсивную функцию вычисления функции Аккермана для всех неотрицательных целых аргументов m и n :

$$A(m, n) = \begin{cases} A(0, n) = n + 1 \\ A(m, 0) = A(m - 1, 1), m > 0 \\ A(m, n) = A(m - 1, A(m, n - 1)), m, n > 0. \end{cases}$$

Вариант 7. Написать рекурсивную функцию вычисления :Для заданных границ интегрирования a и b вычислите значение определенного интеграла следующего вида:

$$\int_a^b \sin^n x = \begin{cases} -\frac{\sin^{n-1} x \cdot \cos x}{n} + \frac{n-1}{n} \cdot \int_a^b \sin^{n-2} x dx, & n > 2 \\ \frac{x}{2} - \frac{1}{4} \cdot \sin x, & n = 2 \\ -\cos x, & n = 1; \end{cases}$$

Вариант 8. Написать рекурсивную функцию вычисления :Пусть даны целое n от 2 до 20 и вещественное $E > 0$. Найдите с точностью E все корни n -го многочлена Чебышева $T_n(x)$, определяемого формулами: $T_0(x) = 1$; $T_1(x) = x$; $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$, $k = 2, 3, \dots$

Вариант 9. Задано вещественное число $a > 0$. Вычислить $\frac{\sqrt[3]{a} - \sqrt[6]{a^2 + 1}}{1 + \sqrt[3]{3 + a}}$.

Вычисление корней $y = \sqrt[k]{x}$ оформить в виде подпрограммы. Корни вычислять с точностью $E = 0.00001$ по итерационной формуле:

$$y_0 = 1; y_{n+1} = y_n + \frac{\left(\frac{x}{y_n^{k-1}} - y_n \right)}{k}, \text{ приняв за ответ приближение, для}$$

которого $|y_{n+1} - y_n| < E$. Вычисление y^k оформить в виде рекурсивной подпрограммы.

Вариант 10. Напишите рекурсивную функцию для нахождения биномиальных коэффициентов (для заданного $M \geq i \geq j > 0$ вычислите все C_i^j).

$$C_n^m = \begin{cases} 1, & \text{при } m = 0, n > 0 \text{ или } n = m \geq 0 \\ 0, & \text{при } m > n > 0 \\ C_{m-1}^{n-1} + C_m^{n-1}, & \text{иначе} \end{cases}.$$

Вариант 11. Написать рекурсивную функцию вычисления *Наибольшего общего делителя* по алгоритму Эйлера:

Если B делится на A нацело, то $\text{НОД}(A, B) = A$.

В противном случае $\text{НОД}(A, B) = \text{НОД}(B \bmod A, A)$.

Вариант 12. Напишите две функции вычисления i -го числа Фибоначчи: рекурсивную и не рекурсивную. Напечатайте таблицу для сравнения времени вычисления i -го числа. Вычислите $f(k)$, $k = 15, 20, 30, 40$. Функция $f(n)$ определена для целых чисел следующим образом:

$$f(n) = \begin{cases} 1, & n = 1 \\ \sum_2^n f(n \operatorname{div} 2), & n \geq 2 \end{cases}.$$

Вариант 13. Для заданных m и x вычислить бином Ньютона $(1+x)^m$ непосредственно и по формуле разложения в ряд: $(1+x)^m = \sum_{i=0}^m C_m^i \cdot x^i$. Для вычисления C_m^i использовать рекуррентное соотношение (написать рекурсивную функцию): $C_m^{i+1} = C_m^i \frac{m-i}{i+1}$; $C_m^0 = 1$.

Вариант 14. Написать рекурсивную функцию вычисления функции Аккермана для всех неотрицательных целых аргументов m и n :

$$A(m, n) = \begin{cases} A(0, n) = n + 1 \\ A(m, 0) = A(m-1, 1), m > 0 \\ A(m, n) = A(m-1, A(m, n-1)), m, n > 0. \end{cases}$$

Библиографический список

- 1) Павловская Т.А. С/С++. Программирование на языке высокого уровня : учеб. для вузов/ Т.А. Павловская. -М.; СПб.: Питер, 2006. -460 с.:а-ил.
- 2) Мейер Б. Основы программирования. ИНТУИТ, 2011. -456 с. ЭБС "КнигаФонд"
- 3) Афонин В.В. Александров Э.Э. Программирование на языке С Microsoft Visual Studio 2010/ ИНТУИТ 2010 г. – 712 с. ЭБС "КнигаФонд"
- 4) Туральчук К.А. Параллельное программирование с помощью языка С# ИНТУИТ 2010 г. – 345 с. ЭБС "КнигаФонд"

Учебное издание

Ведерникова Ольга Геннадьевна

ПРОГРАММИРОВАНИЕ

Часть 2

**Средства построения многомодульных программ с
использованием функций**

«Электронный университет» ФГБОУ ВО РГУПС

Адрес университета:
344038, Ростов н/Д, пл. Ростовского Стрелкового Полка
Народного Ополчения, 2.