

Содержание

Содержание	1
Содержание отчёта по лабораторной работе	3
Лабораторная работа № 1	3
Теоретические сведения	3
Паскаль	3
Математические операторы	3
Операторы ввода и вывода	4
C++	4
Математические операторы	4
Операторы ввода и вывода	5
Варианты заданий	5
Пример выполнения лабораторной работы	6
Лабораторная работа № 2	8
Теоретические сведения	8
Паскаль	8
C++	8
Варианты заданий	8
Лабораторная работа № 3	9
Теоретические сведения	10
Паскаль	10
Операторы цикла с параметром (for)	10
Оператор цикла с постусловием (repeat)	10
Операторы цикла с предусловием (while)	10
C++	11
Цикл FOR	11
Цикл while	11
Цикл DO	11
Варианты заданий	11
Лабораторная работа № 4	12
Теоретические сведения	12
Паскаль	12
C++	12
Варианты заданий	12
Лабораторная работа № 5	14
Теоретические сведения	14
Паскаль	14
Ввод массива	14
Вывод массива	14
C++	14
Ввод массива	15
Вывод массива	15
Варианты заданий	15
Лабораторная работа № 6	16
Теоретические сведения	16
Паскаль	16
Ввод/вывод двумерных массивов:	17
C++	17
Ввод двумерного массива	17
Вывод двумерного массива	17

Варианты заданий.....	17
Лабораторная работа № 7	18
Теоретические сведения.....	18
<i>Паскаль</i>	18
C++	19
C++ автоматически добавляет NULL к строковым константам.....	19
Варианты заданий.....	19
Лабораторная работа № 8	20
Теоретические сведения.....	20
<i>Паскаль</i>	20
C++	21
Варианты заданий.....	22
Лабораторная работа № 9	23
Теоретические сведения.....	23
<i>Паскаль</i>	23
C++	25
Варианты заданий.....	26
Лабораторная работа № 9	26
Теоретические сведения.....	27
<i>Паскаль</i>	27
C++	28
Варианты заданий.....	28

Содержание отчёта по лабораторной работе

В отчёт входит описание:

- постановки задачи;
- исходных данных;
- схема алгоритма;
- текст программы с комментариями;
- результаты работы программы;

Отчёт оформляется на листах формата А4 с обязательным титульным листом, на котором указываются названием работы; ФИО исполнителя; ФИО преподавателя и т.д.

Постановка задачи

Постановка задачи указывает, какая цель должна быть достигнута при разработке программы. Какую задачу должна решать программа, и в каких условиях будет функционировать.

Исходные данные

Исходными данными являются любые данные, которые программа получает для обработки.

Описание исходных данных должно содержать:

- семантику (назначение) данных;
- представление в программе.

Результаты работы программы

Необходимо указать, какие результаты производит программа.

Необходимо указать в каком формате пользователь получает результат.

Лабораторная работа № 1

Тема. Расчет простейшей математической формулы

Цель. Научиться использовать математические операторы языков программирования Паскаль и C++. Закрепить навыки использования операторов ввода вывода.

Задание. Написать программу для решения поставленной задачи. Исходные значения вводятся пользователем с клавиатуры, результат расчета выводится на экран.

Теоретические сведения

Паскаль

Математические операторы

Математический оператор	Паскаль
+	+
-	-
*	*
/	/
$\sin a$	sin(a)
$\cos a$	cos(a)
$\log_a b$	ln(b)/ln(a)
$\ln a$	ln(a)
\sqrt{a}	sqrt(a)
a^2	sqr(a)
a^x	exp(x*ln(a))
$\lg b$	ln(b)/ln(10)
$ a $	abs(a)
целочисленный деление a на b	a div b
получение остатка от деления a на b	a mod b
округление до ближайшего целого числа x	round(x)
обрезка до целого числа x	trunc(x)
возврат дробной части числа x	frac(x)

возврат целой части числа x

int(x)

Операторы ввода и вывода

Для ввода исходных данных используются операторы процедур ввода:

Read(A1,A2,...AK);

ReadLn(A1,A2,...AK);

ReadLn;

Первый из них реализует чтение K значений исходных данных и присваивание этих значений переменным A1, A2, ..., AK. Второй оператор реализует чтение K значений исходных данных, пропуск остальных значений до начала следующей строки, присваивание считанных значений переменным A1, A2, ..., AK. Третий оператор реализует пропуск строки исходных данных. При вводе исходных данных происходит преобразование из внешней формы представления во внутреннюю, определяемую типом переменных. Переменные, образующие список ввода, могут принадлежать либо к целому, либо к действительному, либо к символьному типам. Чтение исходных данных логического типа в языке ПАСКАЛЬ недопустимо.

Для вывода результатов работы программы на экран используются операторы:

Write(A1,A2,...AK);

WriteLn(A1,A2,...AK);

WriteLn;

Первый из этих операторов реализует вывод значений переменных A1,A2,...,AK в строку экрана. Второй оператор реализует вывод значений переменных A1, A2, ..., AK и переход к началу следующей строки. Третий оператор реализует пропуск строки и переход к началу следующей строки. Переменные, составляющие список вывода, могут относиться к целому, действительному, символьному или булевскому типам. В качестве элемента списка вывода кроме имен переменных могут использоваться выражения и строки. Вывод каждого значения в строку экрана происходит в соответствии с шириной поля вывода, определяемой конкретной реализацией языка. Форма представления значений в поле вывода соответствует типу переменных и выражений: величины целого типа выводятся как целые десятичные числа, действительного типа - как действительные десятичные числа с десятичным порядком, символьного типа и строки - в виде символов, логического типа - в виде логических констант TRUE и FALSE.

Оператор вывода позволяет задать ширину поля вывода для каждого элемента списка вывода. В этом случае элемент списка вывода имеет вид A:K, где A - выражение или строка, K - выражение либо константа целого типа.

Если выводимое значение занимает в поле вывода меньше позиций, чем K, то перед этим значением располагаются пробелы. Если выводимое значение не помещается в ширину поля K, то для этого значения будет отведено необходимое количество позиций. Для величин действительного типа элемент списка вывода может иметь вид A:K:M, где A - переменная или выражение действительного типа, K - ширина поля вывода, M - число цифр дробной части выводимого значения. K и M - выражения или константы целого типа. В этом случае действительные значения выводятся в форме десятичного числа с фиксированной точкой.

C++

Математические операторы

Основные математические операторы

Математический
оператор

+

-

*

/

++

--

Паскаль

+

-

*

/

Инкремент

Декремент

Математические функции заголовочного файла math.h

sin a

cos a

arctg x

sin(a)

cos(a)

atan(x)

$\log_a b$	$\log(b)/\log(a)$
$\ln a$	$\log(a)$
\sqrt{a}	$\text{sqrt}(a)$
a^x	$\text{pow}(a,x)$
$\lg b$	$\log(b)/\log(10)$
$ a $	$\text{abs}(a)$
получение остатка от деления a на b	$a \% b$

Операторы ввода и вывода

Для вывода символов и чисел на экран вы можете использовать выходной поток *cout*.

- В C++ можно использовать с *cout* специальные символы для вывода табуляции или новой строки и даже для воспроизведения звука на вашем компьютере.
- В C++ можно легко отображать числа в десятичном, восьмеричном (по основанию 8) или шестнадцатеричном (по основанию 16) формате.
- Используя в командной строке операционной системы операторы переназначения, вы можете перенаправить выходные сообщения своей программы, посылаемые в *cout*, с экрана в файл или на принтер.
- Используя выходной поток *cerr*, ваши программы могут посылать сообщения на стандартное устройство ошибок, избавляя пользователей от необходимости переназначения сообщений.

Для управления шириной вывода вещественных чисел используется функция-прототип *precision*(количество знаков после запятой).

Пример вывода значений.

```
#include <iostream.h>
void main(void)
{ int i=100;
  float j=1.33;
  cout.precision(3);
  cout << "i ="<i><<"j = "<i><<j;
}
```

Для чтения ввода с клавиатуры программы могут использовать входной поток *cin*. При использовании *cin* вы должны указать переменную, в которую *cin* помещает данные. Затем используйте оператор извлечения (>>) для направления данных, как показано ниже: *cin >> список ввода*;

Оператор извлечения называется так, потому что он извлекает (удаляет) данные из входного потока, присваивая значение указанной переменной.

Варианты заданий

1. Всхожесть семян пшеницы равна 95%. Сколько тонн семян нужно посеять, чтобы 35 тонн из них проросло?
2. Расстояние между домами, в которых живут Петя и Коля, 1200 м. Однажды они вышли каждый из своего дома и направились навстречу друг другу. Когда Петя прошел А метров, они встретились. Во сколько раз расстояние, которое прошел Коля больше расстояния, которое прошел Петя?
3. От станции до озера S км. Туристы, направляясь от станции к озеру, полтора часа шли пешком со скоростью А км, а затем сели на попутную машину, которая ехала со скоростью В км/ч. За какое время туристы добрались до озера?
4. Из 18 т железной руды выплавляют 10 т железа. Сколько железа выплавят из А т руды?
5. Чтобы заварить 1,5 л чая, нужно 30 г сухого чая. Чайник вмещает В л. Сколько нужно сухого чая для заварки?
6. В школьном коридоре длиной 56 м нужно выкрасить пол. Выкрасив часть коридора длиной 22 м, израсходовали А кг краски. Сколько еще нужно краски, докрасить коридор?

7. Чтобы сварить 4 порции пшенной каши, нужно взять 220 г пшена, 1 л молока и 30 г сахара. Сколько потребуется этих продуктов, чтобы сварить N порций каши?
8. Универмаг в ноябре продал товаров на A млн. р., что на 15% больше, чем в октябре. На какую сумму были проданы товары в октябре?
9. Сахарная свекла содержит 14% сахара. С 1 га собирают 30 т сахарной свеклы. сколько гектаров надо засеять сахарной свеклой, чтобы получить B т сахара?
10. Клоун предложил каждому из публики задумать число. Потом он сказал: «Прибавьте к задуманному числу 5. Теперь из результата вычтите 2. А теперь к результату прибавьте 7». Потом клоун спросил у желающих, какое число у каждого из них получилось. Услышав ответ, он немедленно объявлял каждому, какое число тот задумывал. Составьте программу, которая повторяла бы фокус клоуна.
11. Для того, чтобы печенье было рассыпчатым, в него кладут пекарский порошок. В его состав входят пищевая сода – 25%, лимонная кислота – 15% и мука – 60%. Сколько соды, лимонной кислоты и муки надо взять, чтобы приготовить A г пекарского порошка?
12. Чтобы связать один шарф, требуется 200 г пряжи. Пряжа состоит на 40% из мохера и на 60% из полиакрила. Сколько мохера и полиакрила потребуется, чтобы изготовить B шарфов?
13. Мельхиор – это сплав никеля и меди, массы которых пропорциональны числам 2 и 9. Сколько потребуется никеля и меди для выплавки C кг мельхиора?
14. Для приготовления квашеной капусты на 10 кг капусты берут 225 г соли, 350 г моркови, 200 г клюквы, 4 г лаврового листа. Сколько соли, моркови, клюквы и лаврового листа надо взять, если закуплено M кг капусты?
15. Для приготовления компота из персиков берут сахар, персики и воду в пропорции 1:1:3. Сколько кг каждого продукта надо взять, чтобы сварить P кг компота?
16. Масса молекул кислорода в организме человека составляет 65% от его массы. Составьте программу для определения массы молекул кислорода в организме человека.
17. В классе N учеников. После контрольной было получено: A - пятерок, B - четверок, C - троек. Найти процент троек, четверок и пятерок.
18. Четырем классам поручено убрать поле площадью F га. Известно количество учеников в каждом классе. Сколько га надо убрать каждому классу?
19. Даны два числа. Найти среднее арифметическое кубов этих чисел и среднее геометрическое модулей этих чисел.
20. Вычислить расстояние между двумя точками с данными координатами x_1, y_1 и x_2, y_2 .
21. По двум данным катетам найти гипотенузу и площадь прямоугольного треугольника. Даны числа x и y. Вычислить их сумму, разность и произведение.
22. Автомобиль проехал три участка пути разной длины с разными скоростями. Найти среднюю скорость автомобиля.
23. Известна сумма денег, имеющаяся у покупателя и стоимость одной единицы товара. Сколько единиц товара сможет купить покупатель и какова его сдача?
24. Составьте программу для вычисления суммы цифр введенного с клавиатуры трехзначного натурального числа.
25. Известен плановый P и фактический F показатели выпуска продукции предприятием. Определить процент выполнения плана.

Пример выполнения лабораторной работы

Задание. Цех по ремонту бытовой техники имеет ежедневно прибыль P, с которой платит налог 70%. Из оставшейся доли прибыли 30% идет на развитие производства, а остальное – на заработную плату 8 работникам. Какова среднемесячная зарплата работника?

Постановка задачи. Необходимо рассчитать зарплату работника формуле:

$$K = (0.3P - 0.3P/3)/8$$

Исходные данные. Исходные данные представлены в виде таблицы.

Переменная	Тип
Входные данные	
Р	вещественный
Выходные данные	
К	вещественный

Схема алгоритма. Схема алгоритма показана на рисунке 1.

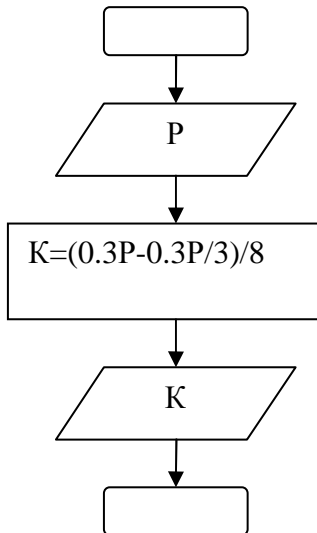


Рисунок 1. Схема алгоритма
Текст программы.

Паскаль

```

Uses crt;
Var
    K,P:real;
BEGIN
    Clrscr; { Очистка экрана }
    Write('P = ');
    Readln(P);
    K:=(0.3*P-0.3*P/3)/8;
    Writeln('K= ',K:5:3);
    Readkey; { Задержка результата на экране до нажатия любой клавиши }
END.
  
```

C++

```

#include <iostream.h>
#include <conio.h>
void main ()
{
    float P,K;
    clrscr(); //Очистка экрана
    cout << "P = ";
    cin >> P;
    K=(0.3*P-0.3*P/3)/8;
    cout.precision(3) ;
    cout << "K = "<<K;
    cout <<"\n"; //переход на новую строку
    getch(); // Задержка результата на экране до нажатия любой клавиши
}
  
```

Лабораторная работа № 2

Тема. Программирование разветвляющихся процессов

Цель. Научиться использовать операторы условия языков программирования Паскаль и C++.

Задание. Написать программу для решения поставленной задачи. Исходные значения вводятся пользователем с клавиатуры, результат расчета выводится на экран.

Теоретические сведения

Паскаль

Условный оператор позволяет выбрать одно из двух действий. Существует два вида условного оператора:

1) *if b then s1;*

2) *if b then s1 else s2 ;*

где *b* - выражение логического типа;

s1, s2-отдельные операторы или операторы, сгруппированные вместе при помощи операторных скобок *begin end*. Такой оператор называется составным.

C++

Условный оператор имеет две формы записи:

if (выражение) оператор1;

if (выражение) оператор1; else оператор2;

Если выражение истинно, то выполняется *оператор1*, если оно ложно, то при использовании *формы 1* управление передается следующему оператору, а при применении *формы 2* выполняется *оператор2*.

Короткий способ записи оператора *if* – условная операция. Форма записи оператора следующая:

выражение1 ? выражение2 : выражение3;

"*Выражение1*" должно быть целого или плавающего типа или указатель. Если "*выражение1*" равно нулю (ложно), то вычисляется "*выражение3*", и его значение является результатом операции. Если значение "*выражения1*" отлично от нуля (истинно), то результатом операции является значение "*выражения2*".

Варианты заданий

1. Заданы числа *a* и *b*. Определить, эти числа одного или разных знаков.
2. Напишите программу-модель анализа пожарного датчика в помещении, которая выводит сообщение «Пожарная ситуация», если температура (ее значение вводится с клавиатуры) в комнате превысила 60°C.
3. Рис расфасован в два пакета. Вес первого - *m* кг, второго – *n* кг. Составьте программу, определяющую: а) какой пакет тяжелее – первый или второй? б) вес более тяжелого пакета.
4. Туристы вышли из леса на шоссе неподалеку от километрового столба с отметкой *A* км и решили пойти на ближайшую автобусную остановку. Посмотрев на план местности, руководитель группы сказал, что автобусные остановки расположены на километре *B* и на километре *C*. Куда следует пойти туристам?
5. Валя и Вера на своем садовом участке собрали *A* кг клубники. Из них *B* кг собрала Вера. Кто из девочек собрал клубники больше и на сколько?
6. Стоимость *A* метров серой ткани равна *B* рублей, а стоимость *K* метров синей ткани равна *M* рублей. Какая ткань дороже и на сколько?
7. Первая бригада маляров за *t1* час покрасила *A* м² стен, а вторая бригада за *t2* часа покрасила *B* м². У какой бригады производительность труда выше и на сколько?
8. Турист за день прошел *A* км. До обеда он шел *t* часов и прошел 20 км. Еще *t2* часа он шел после обеда. Когда скорость туриста была выше: до обеда или после обеда?

9. Тело имеет массу M г и объем V см³. Будет ли оно плавать в жидкости, объем которой равен V_1 см³, а масса M_1 г? (Указание: найдите плотности тела и жидкости, а затем сравните их).
10. Валя и Вера собрались варить варенье из A кг смородины. По рецепту на 2 кг ягод нужно 3 кг сахара. Валя сказала, что им потребуется C кг сахара, а Вера - что P кг. Кто из них прав?
11. Масса 8 литров бензина 5,68 кг. Цистерна имеет объем 500 м³. Хватит ли ее, чтобы вместить A т бензина?
12. Кусок медного провода длиной 5 м имеет массу 430 г. чтобы провести проводку в квартире требуется C метров. Хватит ли для этой цели мотка провода массой M г?
13. Смекалкин, уходя в школу, вышел из дому на 3 минуты позже младшего брата. Расстояние до школы S метров. Смекалкин идет со скоростью V_1 м/мин, а его брат – V_2 м/мин. Догонит ли Смекалкин брата, прежде, чем тот придет в школу?
14. Составьте программу, проверяющую, верно ли утверждение, что введенное вами целое число является четным.
15. Составьте программу, проверяющую, верно ли утверждение, что введенное вами целое число делится без остатка на 3.
16. Составьте программу, проверяющую, верно ли утверждение, что сумма цифр введенного вами целого числа является четным.
17. Составьте программу, проверяющую, верно ли утверждение, что сумма цифр введенного вами целого числа делится на 5.
18. Составьте программу, определяющую, пройдет ли график функции $y = 5x^2 - 7x + 2$ через заданную точку с координатами (a, b) .
19. Даны три действительных числа a, b, c . Найти наибольшее из них.
20. Фермер намерен купить корову, дающую не менее L литров молока ежедневно с жирностью не менее K процентов. Написать алгоритм и программу выбора коровы.
21. Написать алгоритм и программу проверки, является ли данный четырехугольник параллелограммом.
22. Гнездо для ножки деревянной кровати имеет форму прямоугольной вырезки. Определить, подойдет ли данный четырехугольный брусок для этого гнезда.
23. Стороны одного прямоугольника равны A и B . Стороны другого равны X и Y . Написать алгоритм и программу проверки прямоугольников на равенство.
24. Занятия в начальных классах отменяются в тех случаях, когда температура воздуха не выше -25 градусов, а также при ветре не менее 7м/с и температуре не выше -20 градусов. По утренней сводке погоды определить, пойдут ли дети в школу.
25. Стороны одного треугольника равны A_1, B_1, C_1 . Написать алгоритм и программу проверки равенства ему другого треугольника.

Лабораторная работа № 3

Тема. Табулирование значений функции

Цель. Научиться использовать операторы циклов языков программирования Паскаль и C++.

Задание. Написать программу для табулирования значений функций на заданном интервале $[x_n, x_k]$ с шагом h . Значения интервала и шаг вводится пользователем, результат выводится в виде таблицы со значением x и y . Также необходимо провести проверку на существование значения y в данной точке.

Для отображения таблицы на экран можно использовать следующие символы псевдографики:

191 \neg 179 | 180 \neg 192 \perp 193 \perp 194 \neg 195 \neg 196 \neg 197 \neg

Для ввода такого символа в Паскале или C++ необходимо зажать клавишу ALT и набрать код на цифровой клавиатуре, после чего отпустить клавишу.

Теоретические сведения

Паскаль

Циклы классифицируются следующим образом:

1. *цикл с предусловием*
2. *цикл с постусловием*
3. *цикл с параметром*.

Оператор цикла задает повторное выполнение определенных операторов. Если число повторений заранее известно, то подходящей конструкцией является оператор *for*. В противном случае следует использовать операторы *while* или *repeat*.

Для управления повторением операторов можно использовать стандартные процедуры *Break* и *Continue*. *Break* завершает оператор цикла, а *Continue* продолжает со следующей итерации этого оператора.

Операторы цикла с параметром (*for*)

Операторы цикла с параметром вызывает повторяющееся выполнение оператора (который может быть составным оператором) пока управляющей переменной присваивается возрастающая последовательность значений.

В качестве управляющей переменной должен использоваться идентификатор переменной, который обозначает переменную, объявленную локальной в блоке, в котором содержится оператор *for*. Управляющая переменная должна иметь перечислимый тип. Начальное и конечное значения должны иметь тип, совместимый по присваиванию с перечислимым типом.

for параметр := начальное значение *to* конечное значение *do* оператор;

Оператор, который содержится в теле оператора *for*, выполняется один раз для каждого значения в диапазоне между начальным и конечным значением. Когда работает оператор *for*, значение управляющей переменной (счетчика циклов) увеличивается при каждом повторении на единицу. Если начальное значение превышает конечное значение, то содержащийся в теле оператора *for* оператор не выполняется. Когда в операторе цикла используется ключевое слово *downto*, значение управляющей переменной уменьшается при каждом повторении на единицу. Если оператор, содержащийся в теле оператора *for*, изменяет значение управляющей переменной, то это является ошибкой.

Оператор цикла с постусловием (*repeat*)

В операторе цикла с постусловием выражение, которое управляет повторным выполнением последовательности операторов содержится внутри оператора *repeat*. Результат выражения должен быть булевского типа.

repeat

оператор1;

оператор2;

...

операторn;

until условие;

Операторы, заключенные между ключевыми словами *repeat* и *until*, выполняются последовательно до тех пор, пока результат выражения не примет значение *True*. Последовательность операторов выполнится по крайней мере один раз.

Операторы цикла с предусловием (*while*)

Оператор цикла с предусловием содержит в себе выражение, которое управляет повторным выполнением оператора (который может быть составным оператором).

while условие *do* оператор

Выражение, с помощью которого осуществляется управление повторением оператора, должно иметь булевский тип. Вычисление его производится до того, как внутренний

оператор будет выполнен. Внутренний оператор выполняется повторно до тех пор, пока выражение принимает значение True.

C++

Цикл FOR

Если ваша программа использует оператор *for* (часто называемый циклом *for*), она должна указать переменную, которая называется *управляющей переменной*, хранящей количество выполнений цикла.

for (начальное значение счетчика; условие работы цикла; шаг)
оператор;

Цикл while

while (условие)

оператор;

Если ваша программа встречает оператор *while*, она проверяет заданное условие. Если условие истинно, программа выполняет операторы цикла *while* до тех пор пока, условие не станет ложным.

Цикл DO

Если программа встречает цикл *do while*, она входит в цикл и запускает выполнение операторов, содержащихся в цикле. Затем программа оценивает заданное условие. Если условие истинно, программа возвращается к началу цикла. Если условие ложно, программа не повторяет инструкции цикла, продолжая вместо этого выполнение с первого оператора, следующего за циклом.

do

{

оператор;

}

while (условию);

Во всех трех циклах *оператор* может быть составным.

Варианты заданий

$$1. Y = \sqrt{|x-a|} * b \sin p$$

$$2. Y = x(\arctg p + a \cos b) / a$$

$$3. Y = \sin^3(ax^2 + bx + p) / (a-x)$$

$$4. Y = (1 + \cos(x-a)) / (2b-p)$$

$$5. Y = \sin^2(\arctg(x/a)) + b * p$$

$$6. Y = (x+a) / (x^2 + b) - \cos p$$

$$7. Y = ax^2 \sin bp / (x+a)$$

$$8. Y = bp^3 \cos ax / (x-a)$$

$$9. Y = \sin \sqrt{a} x + \cos bp$$

$$10. Y = a \sin^2(3x-b) / p$$

$$11. Y = -a + x^3 (\cos b + \sin p) / a$$

$$12. Y = \sqrt{(ax^3 + bx^2 + p)^2 + 3.14} / x$$

$$13. Y = \sin^3(ax^2 + bx + p) / (a-x)$$

$$14. Y = (1 + \cos(x-a)) / (2b-p)$$

$$15. Y = \sin^2(\arctg(x/a)) + b * p$$

$$16. Y = (x+a) / (x^2 + b) - \cos p$$

$$17. Y = ax^2 \sin bp / (a-x)$$

18. $Y = ax \cos bp^3 / (a-x)$
19. $Y = \sqrt{ax} + \operatorname{tg} bp$
20. $Y = (3x-b) \sin^2 p / a$
21. $Y = -a \sin p + x^3 \cos b / (a-x)$
22. $Y = \sqrt{(ax^3 + p)^2 + 3.14} + bx^2 / (a-x)$
23. $Y = \sqrt{|x-a|} / b \sin p$
24. $Y = x(\operatorname{ctg} p + a \sin b) / (a-x)$
25. $Y = 15 \cos^2(a * p) / (x+b)$

Лабораторная работа № 4

Тема. Работа с операторами выбора

Цель. Научиться использовать операторы выбора языков программирования Паскаль и C++.

Задание. Написать программу для решения поставленной задачи. Данные вводятся с клавиатуры, а результат выводится на экран.

Теоретические сведения

Паскаль

В общем случае оператор выбора CASE имеет вид:

```
case <селектор> of
  <метка-1>: <оператор-1>;
  <метка-2>: <оператор-2>;
  .....
  <метка-N>: <оператор-N>
else <оператор-(n+1)>;
end;
```

где СЕЛЕКТОР- выражение любого типа, кроме вещественного и строкового;

ОПЕРАТОР - любой оператор языка, в том числе и составной;

МЕТКА - список значений выражения СЕЛЕКТОР или одно его значение.

C++

Общий вид оператора переключателя:

```
switch(выражение) {
  case метка1: операторы1;
  case метка2: операторы2;
  ...
  case меткаN: операторыN;
  default: операторы;
}
```

Значение "выражения" вычисляется и сравнивается с "метками" (обычно это целые или символьные константы). В случае совпадения выполняется группа операторов соответствующая метке. Оператор *default* выполняется, если ни один из предыдущих операторов не выполнялся. Возможно использование нескольких меток перед группой операторов. Наличие ветви *default* необязательно. Желательно в конце группы операторов, соответствующих каждой метке, использовать оператор *break* для завершения выполнения оператора переключателя. В случае отсутствия оператора разрыва сравнение по меткам будет продолжено.

Варианты заданий

1. Составить программу, которая по номеру квартиры выдает фамилию ее владельца.

- 2.Описать список времен года: лето, осень, зима, весна. По введенному значению времени года перечисляла все месяца этого сезона.
- 3.Составить программу, которая бы по названию месяца выдавала бы количество дней в месяце.
- 4.Составить программу, которая бы по названию месяца выдавала бы время года, к которому он принадлежит.
- 5.Составить программу, которая бы по порядковому номеру месяца выдавала бы его название.
- 6.Составить программу, которая бы по порядковому номеру месяца выдавала к какому времени года он принадлежит.
- 7.Составить программу, которая бы по введенному времени года выдавала бы название месяцев, относящихся к нему.
- 8.Составить программу, которая бы по названию месяца выдавала бы его порядковый номер и название времени года.
- 9.Дан список дисциплин и отчетность по ним. Составить программу, которая бы по названию дисциплины выдавала бы отчетность по нему.

История (экзамен, зачет)

Культурология (зачет)

Философия (экзамен, зачет)

Иностранный язык (экзамен, зачет)

Экономика (экзамен)

- 10.Дан список дисциплин и номер семестра, когда они изучаются. Составить программу, которая бы по номеру семестра выдавала бы список изучаемых дисциплин.

История-2,1

Культурология-3,4

Философия-4,3

Ин.язык-4,1,2,3

- 11.По списку дисциплин приведенных в 10 варианте заданий составить программу, которая выдавала бы список дисциплин, читаемых на определенном курсе. Учитывать, что 1 курс это 1 и 2 семестр, 2 курс-3,4 семестр и т.д.
- 12.Составить программу, которая бы с помощью оператора CASE реализовала бы все возможные операции над двумя целыми числами.
- 13.Составить программу, которая бы с помощью оператора CASE реализовала бы все возможные операции над вещественными числами.
- 14.Составить программу, которая бы присваивала переменной T значение true, если дата d1,m1 предшествует(в рамках года) дате d2,m2 и значение false иначе(d1 и d2-дата,m1 и m2-месяц).
- 15.Составить программу, которая бы выдавала название месяца, следующего за введенным месяцем (с учетом того, что за декабрем идет январь).
- 16.Составить программу, которая бы выдавала по названию страны название столицы этой страны (использовать не менее 6-7 названий).
- 17.Составить программу, которая бы по первой букве в названии языка программирования выводила английское название этого языка.
- 18.Составить программу, которая бы по введенному числу (до 10) выдавала бы название этой цифры.
- 19.Составить программу, которая бы по введенному названию страны выдавала название ее континента.
- 20.Составить программу, которая бы по значению переменной X, означающему некоторую длину в следующих единицах измерения: дециметр, километр, метр, миллиметр, сантиметр; выдавала бы эту длину в метрах.
- 21.Составить программу, которая реализовала бы следующие действия: по введенному числу K (до 10)выдавала бы соответствующую ей римскую цифру.

22. Для целого числа K от 1 до 9 напечатать фразу "мне K лет", учитывая при этом, что при некоторых значениях K слово "лет" надо заменить на слово "год" или "года"
23. Для натурального числа K напечатать фразу "мы нашли K грибов в лесу", согласовав окончание слова "гриб" с числом K.
24. Составить программу, которая бы реализовала следующий алгоритм: переменной T присвоить значение true если сочетание день.месяц образует правильную дату, и значение false- иначе(учитывая количество дней в месяце и название месяца).
25. Составить программу, которая бы реализовала следующий алгоритм: по порядковому номеру дня года определить дату, т.е. число и месяц.

Лабораторная работа № 5

Тема. Работа с одномерными массивами

Цель. Научиться использовать одномерные массивы в программах Паскаль и C++.

Задание. Написать программу для обработки одномерного массива. Данные вводятся с клавиатуры, а результат выводится на экран.

Теоретические сведения

Паскаль

Массивы представляют собой ограниченную упорядоченную совокупность однотипных величин. Каждая отдельная величина называется компонентой массива. Тип компонент может быть любым, принятым в языке ПАСКАЛЬ, кроме файлового типа. Тип компонент называется базовым типом.

Вся совокупность компонент определяется одним именем. Для обозначения отдельных компонент используется конструкция, называемая переменной с индексом или с индексами. Массив объявляется следующим образом:

имя массива: array [тип индекса] of тип элементов;

В качестве индекса может быть использовано выражение. Тип индексов может быть только интервальным или перечисляемым. Действительный и целый типы недопустимы. Индексы интервального типа, для которого базовым является целый тип, могут принимать отрицательные, нулевое и положительные значения. В операторной части программы один массив может быть присвоен другому, если их типы идентичны, например:

R1:=Z.

Для ввода или вывода массива в список ввода или вывода помещается переменная с индексом, а операторы ввода или вывода выполняются в цикле.

Ввод массива

```
for i:=1 to n do begin
    write('a[',i,']=');
    readln(a[i]);
end;
```

Вывод массива

```
for i:=1 to n do
    write('a[',i,']= ',a[i]);
```

C++

Массив представляет собой переменную, способную хранить одно или несколько значений. Подобно переменным, используемым вашими программами до сих пор, массив должен иметь тип (например, *int*, *char* или *float*) и уникальное имя.

Тип_массива имя_массива [размерность]

Первый элемент массива всегда имеет индекс 0, а значение индекса последнего элемента на единицу меньше размера массива.

При объявлении массива можно указать первоначальные значения, поместив их

между левой и правой фигурными скобками, следующими за знаком равенства. Например, следующий оператор инициализирует массив *values*:

```
int values[5] = { 100, 200, 300, 400, 500 };
```

Если вы не указываете первоначальное значение для какого-либо элемента массива, большинство компиляторов C++ будут инициализировать такой элемент нулем. Например, следующее объявление инициализирует первые три из пяти элементов массива:

```
int values[5] = { 100, 200, 300 };
```

Программа не инициализирует элементы *values[3]* и *values[4]*. В зависимости от вашего компилятора, эти элементы могут содержать значение 0. Если вы не указываете размер массива, который вы инициализируете при объявлении, C++ распределит достаточно памяти, чтобы вместить все определяемые элементы. Например, следующее объявление создает массив, способный хранить четыре целочисленных значения:

```
int numbers[] = { 1, 2, 3, 4 };
```

Ваши программы будут передавать массивы в функции точно так же, как и любые другие переменные. Функция может инициализировать массив, прибавить к массиву значения или вывести элементы массива на экран. Когда вы передаете массив в функцию, вы должны указать тип массива. Нет необходимости указывать размер массива.

Ввод массива

```
for (i=0;i<n;i++)  
{  
    cout<<"a["<<i<<"] =";  
    cin>>a[i];  
}
```

Вывод массива

```
for (i=0;i<n;i++)  
    cout<<"a["<<i<<"] = "<<a[i]<<"\n";
```

Варианты заданий

1. Введите с клавиатуры в массив пять целочисленных значений. Выведите их в одну строку через запятую. Получите для массива среднее арифметическое.
2. Введите с клавиатуры пять целочисленных элементов массива X. Выведите на экран значения корней и квадратов каждого из элементов массива.
3. Создайте массив из пяти фамилий и выведите их на экран столбиком, начиная с последней.
4. Создайте массив из пяти фамилий и выведите их на экран те из них, которые начинаются с определенной буквы, которая вводится с клавиатуры.
5. Создайте массив A[1..7] с помощью генератора случайных чисел и выведите его на экран. Увеличьте все его элементы в 2 раза.
6. Создайте массив A[1..8] с помощью генератора случайных чисел с элементами от -10 до 10 и выведите его на экран. Подсчитайте количество отрицательных элементов массива.
7. Создайте массив A[1..12] с помощью генератора случайных чисел с элементами от -20 до 10 и выведите его на экран. Замените все отрицательные элементы массива числом 0.
8. Создайте целочисленный массив A[1..15] с помощью генератора случайных чисел с элементами от -15 до 30 и выведите его на экран. Определите самый большой элемент массива и его индекс.

9. С 8 до 20 часов температура воздуха измерялась ежечасно. Известно, что в течение этого времени температура понижалась. Определите, в котором часу была впервые отмечена отрицательная температура.
10. Данные о температуре воздуха за декаду ноября хранятся в массиве. Определить, сколько раз температура опускалась ниже -10 градусов.
11. Данные о температуре воды на Черноморском побережье за декаду сентября хранятся в массиве. Определить, сколько за это время было дней, пригодных для купания.
12. Данные о температуре воздуха и количестве осадков за декаду апреля хранятся в массивах. Определить количество осадков, выпавших в виде дождя и в виде снега за эту декаду.
13. Данные о температуре воздуха за декаду декабря хранятся в массиве. Определить, сколько раз температура была выше средней за эту декаду..
14. Данные о направлении ветра (северный, южный, восточный, западный) и силе ветра за декаду ноября хранятся в массиве. Определить, сколько дней дул южный ветер с силой, превышающей 8 м/с.
15. Создайте массив из 15 целочисленных элементов и определите среди них минимальное значение.
16. Сформируйте линейный массив вещественных чисел, элементы которого являются расстояниями, пройденными телом при свободном падении на землю за $1, 2, \dots, 10$ с.
17. Дан линейный массив целых чисел. Проверьте, является ли он упорядоченным по убыванию.
18. Найти сумму положительных элементов линейного массива целых чисел. Размерность массива – 10. Заполнение массива осуществить с клавиатуры.
19. Найти сумму четных элементов массива целых чисел. Размерность массива – 20. Заполнение массива осуществить случайными числами от 100 до 200.
20. Найти произведение элементов массива целых чисел, которые кратны 7. Размерность массива – 15. Заполнение массива осуществить случайными числами от 10 до 50.
21. Найти сумму элементов массива вещественных чисел, имеющих нечетные номера. Размерность массива – 20. Заполнение массива осуществить случайными числами от 100 до 200.
22. Найти произведение всех элементов массива целых чисел, меньших 0. Размерность массива – 10. Заполнение массива осуществить с клавиатуры.
23. Найти сумму всех элементов массива целых чисел, удовлетворяющих условию: остаток от деления на 2 равен 3. Размерность массива – 20. Заполнение массива осуществить случайными числами от 200 до 300.
24. Найти сумму всех элементов массива вещественных чисел, больших заданного числа. Размерность массива – 20. Заполнение массива осуществить случайными числами от 50 до 100.
25. Найти произведение всех элементов массива вещественных чисел, меньших заданного числа. Размерность массива – 10. Заполнение массива осуществить случайными числами от 50 до 100.

Лабораторная работа № 6

Тема. Работа с двумерными массивами

Цель. Научиться использовать двумерные массивы в программах Паскаль и C++.

Задание. Написать программу для обработки двумерного массива. Данные вводятся с клавиатуры, а результат выводится на экран.

Теоретические сведения

Паскаль

Двумерный массив – это таблица каждый элемент которой имеет две координаты: номер строки и номер столбца. Он объявляется следующим образом:

имя массива: `array [тип индекса1, тип индекса2] of тип элементов;`

Ввод/вывод двумерных массивов:

Ввод

```
for i:=1 to n do
for j:=1 to m do begin
    write('a[' ,i,j,']=');
    readln(a[i,j]);
end;
```

Вывод

```
for i:=1 to n do
for j:=1 to m do write('a[' ,i,j,']= ',a[i,j]);
```

Первый индекс определяет номер строки, второй - номер столбца. Двумерные массивы хранятся в памяти ЭВМ по строкам.

C++

Двумерный массив в C++ объявляется следующим образом:

Тип_массива имя_массива [размерность] [размерность];

Ввод двумерного массива

```
for (i=0;i<n;i++)
for (j=0;j<n;j++)

{
    cout<<"a["<<i<<j<<"] =";
    cin>>a[i][j];
}
```

Вывод двумерного массива

```
for (i=0;i<n;i++)
for (j=0;j<n;j++)
    cout<<"a["<<i<<j<<"] = "<<a[i][j]<<"\n";
```

Варианты заданий

1. Дан целочисленный массив X[1..5, 1..4]. Заменить в нем все элементы, меньшие 5 числом 111.
2. Дан целочисленный массив B[1..4, 1..3]. Заменить все его элементы их квадратами.
3. Дан действительный массив A[1..5, 1..3]. Напечатать индексы его отрицательных элементов.
4. Создайте двумерный целочисленный массив A[1..10, 1..7] и найдите сумму всех его элементов, имеющих оба нечетных индекса.
5. Дан целочисленный массив A[1..5, 1..3]. Вывести на экран элементы массива, кратные 5, и их индексы.
6. Введите с клавиатуры целочисленные элементы матрицы 3*3 и вычислите сумму элементов каждого столбца.
7. Дан целочисленный массив B[1..5, 1..5]. Напечатать его левую и правую диагонали.
8. Дан целочисленный массив B[1..5, 1..5]. Найти сумму элементов его правой диагонали и произведение элементов левой диагонали.
9. Дан целочисленный массив B[1..5, 1..5]. Найти наибольший элемент массива и сообщить его расположение относительно левой диагонали.
10. Дан целочисленный массив B[1..5, 1..5]. Найти наименьший элемент массива и сообщить его расположение относительно правой диагонали.
11. Дан целочисленный массив B[1..5, 1..5]. Вычислить сумму элементов этого массива, расположенных выше левой диагонали.

12. Дан целочисленный массив $B[1..5, 1..5]$. Вычислить произведение элементов этого массива, расположенных ниже левой диагонали.
13. Дан целочисленный массив $B[1..5, 1..5]$. Найти количество отрицательных элементов этого массива, расположенных ниже левой диагонали.
14. Дан целочисленный массив $B[1..5, 1..5]$. Найти количество положительных элементов этого массива, расположенных выше левой диагонали.
15. Дана таблица со сведениями о наличии сырья на каждом из 4-х участков пяти цехов. Определить номер цеха, в котором меньше всего сырья.
16. Дан массив $A[1..3, 1..15]$. Известно, что среди его элементов два и только два равны между собой. Напечатать их индексы.
17. Даны целые числа a_1, a_2, a_3 . Получить целочисленный массив $B[1..3, 1..3]$, для которого $B[i,j] = a_i - 3a_j$.
18. Даны действительные числа $a_1, a_2, \dots, a_{10}, b_1, b_2, \dots, b_{20}$. Получить действительный массив $C[1..20, 1..10]$, для которого $C[i,j] = a_j / (1 + |b_i|)$.
19. Получить целочисленный массив $A[1..10, 1..12]$, для которого $a_{ij} = i + 2j$.
20. Создайте матрицу 5×5 , значение каждого элемента которой равно сумме номера строки и столбца, на пересечении которых он находится, и вычислите сумму элементов каждой строки.
21. Получить действительный массив $A[1..7, 1..7]$, первая строка которой задается формулой $a_{1j} = 2j + 3$, вторая строка задается формулой $a_{2j} = j + 3 / (2 + j)$, а каждая следующая строка есть сумма двух предыдущих.
22. Дано натуральное число n . Выяснить, сколько положительных элементов содержит массив $A[1..n, 1..n]$, если $a_{ij} = \sin(i + j/2)$.
23. Дан целочисленный массив $A[1..4, 1..5]$. Найти среднее арифметическое каждого из столбцов.
24. Дан действительный массив размерностью $n \times m$, в которой не все элементы равны нулю. Получить новый массив путем деления всех элементов данного массива на ее больший по модулю элемент.
25. Дан целочисленный массив $A[1..4, 1..5]$. Требуется преобразовать матрицу: поэлементно вычесть последнюю строку из всех строк, кроме последней.

Лабораторная работа № 7

Тема. Работа со строками.

Цель. Научиться обрабатывать строки в программах Паскаль и C++.

Задание. Написать программу для обработки строк. Строка вводится с клавиатуры, а результат выводится на экран.

Теоретические сведения

Паскаль

Значением строкового типа является последовательность символов с динамическим атрибутом длины (в зависимости от действительного числа символов при выполнении программы) и постоянным атрибутом размера в диапазоне от 1 до 255. Для описания строк введен тип *String*. Также строки можно представить в виде массивов символов.

Текущее значение атрибута длины можно получить с помощью стандартной функции *Length*. Отношение между любыми двумя строковыми значениями устанавливается согласно отношению порядка между значениями символов в соответствующих позициях. В двух строках разной длины каждый символ более длинной строки без соответствующего символа в более короткой строке принимает значение "больше"; например, 'Xs' больше, чем 'X'. Нулевые строки могут быть равны только другим нулевым строкам, и они являются наименьшими строковыми значениями.

К символам в строках можно обращаться как к элементам массива.

Следующие процедуры и функции используются для работы со строками Паскаля. *concat (s1,s2,...)*-функция, соединяющая несколько строк в одну

`copy(s,i,c)` – функция, возвращает копию подстроки из строки `s`, начиная с позиции `i`, количество символов – `c`.

`delete(s,i,c)` – процедура удаляет из строки `s`, начиная с позиции `i`, `c` символов.

`insert(s1,s,i)` – процедура вставляет в строку `s` подстроку `s1`, начиная с подстроки `i`.

`length(s)` – функция возвращает длину строки.

`pos(s1,s)` – функция возвращает позицию первого вхождения подстроки `s1` в строку `s`, если подстрока не найдена возвращает 0.

`str(x,s)` – процедура, преобразует число `x` в строку.

`val(s,x,c)` – процедура, преобразует строку в число `x`, если преобразование не удалось переменная `c` содержит 0.

C++

При работе со строками используются следующие концепции:

- Чтобы объявить символьную строку, вы должны объявить массив типа *char*,
- Чтобы присвоить символы символьной строке, ваши программы просто присваивают символы элементам массива символьных строк.
- Программы C++ используют символ NULL (ASCII 0), чтобы отметить последний символ строки. В программах обозначается как `'\0'`.
- C++ позволяет вашим программам инициализировать символьные строки при их объявлении.
- Программы могут передавать символьные строки в функцию, как и любой массив.
- Большинство библиотек этапа выполнения C++ обеспечивают набор функций, которые управляют символьными строками.

Для объявления символьной строки внутри программы просто объявите массив типа *char* с количеством элементов, достаточным для хранения требуемых символов. Например, следующее объявление создает переменную символьной строки с именем *filename*, способную хранить 64 символа (не забывайте, что символ NULL является одним из этих 64 символов):

```
char filename[64];
```

C++ автоматически добавляет NULL к строковым константам

Когда ваши программы выводят символьные строковые константы с помощью выходного потока *cout*, *cout* использует символ NULL (который компилятор добавляет к строке) для определения последнего символа вывода.

При вводе строки компилятор автоматически добавляет символ конца строки, например:

```
cin>>s;
```

Передача символьной строки в функцию подобна передаче любого массива в качестве параметра. Внутри функции вам нужно просто указать тип массива (*char*) и левую и правую скобки массива. Вам не надо указывать размер строки.

Следующий цикл *for* иллюстрирует, как программа может искать NULL в строке:

```
for (index = 0; string[index] != NULL; index++)
```

Поскольку символ NULL равен 0, многие программы упрощают циклы, которые ищут NULL, как показано ниже:

```
for (index = 0; string[index]; index++);
```

В данном случае пока символ, содержащийся в *string[index]* не NULL (0 или ложь), цикл продолжается.

Варианты заданий

1. Напишите программу, которая вводит строку и выводит ее, сокращая каждый раз на 1 символ до тех пор, пока в строке не останется 1 символ.
2. Напишите программу, определяющую число слов в строке. Одно слово от другого отделяется одним пробелом.

3. Напишите программу, которая подсчитывает стоимость телеграммы, текст которой вводится с клавиатуры.
4. Напишите программу, определяющую, какая из букв первая или последняя встречаются в заданном слове чаще.
5. Задано существительное первого склонения, оканчивающееся на букву «а». Напечатайте это слово во всех падежах.
6. Напишите программу, определяющую, сколько букв «о» стоит в заданном слове на четных местах.
7. Заданы фамилия, имя и отчество учащегося, разделенные пробелами. Напишите программу, печатающую фамилию ученика и его инициалы.
8. Напишите программу, удаляющую из слова *i*-ю букву.
9. Напишите программу, удаляющую из текста все буквы, совпадающие с его последней буквой.
10. Напишите программу, удаляющую из слова *X* все буквы, которые встречаются в слове *Z*.
11. Напишите программу, подсчитывающую количество различных букв в заданном слове.
12. Напишите программу, удаляющую из слова каждую третью букву.
13. Напишите программу, которая в заданном тексте меняет слово «Петр» на слово «Иван».
14. Напишите программу, которая удаляет из введенной строки любой требуемый введенный с клавиатуры символ. Процесс удаления выделите в отдельную процедуру.
15. В заданном тексте заменить все буквы "а" на букву "о".
16. Составить программу, которая обращает введенное слово, т.е. переставляет символы в обратном порядке.
17. Составить программу, определяющую, является ли введенное слово перевертышем.
18. Напишите программу, сортирующую символы введенной с клавиатуры строки в порядке возрастания их номеров в ASCII-таблице.
19. Напишите программу, вычисляющую длину самого короткого слова в предложении из трех слов, разделенных пробелами.
20. Составить программу, которая выводит на экран бегущую строку.
21. Составить программу, удаляющую в строке все, что заключено между фигурными скобками и их самих.
22. Написать программу для удаления из строки всех символов * и замены к на а.
23. Написать программу для вставки после числовых значений слова «число», а после символьных слова «символ».
24. Задана строка символов. Необходимо вывести все символы, стоящие между первыми ":" и вторыми ":" .

Лабораторная работа № 8

Тема. Создание собственных процедур и функций.

Цель. Научиться создавать процедуры и функции в программах на языках Паскаль и C++.

Задание. Написать функцию для расчета математической формулы, согласно варианту и процедуры для ввода и вывода данных.

Теоретические сведения

Паскаль

Процедуры и функции представляют собой относительно самостоятельные фрагменты программы, оформленные особым образом и снабженные именем. Упоминание этого имени в тексте программы называется вызовом процедуры (функции).

Отличие функции от процедуры заключается в том, что результатом исполнения операторов, образующих тело функции, всегда является некоторое единственное значение или указатель, поэтому обращение к функции можно использовать в соответствующих

выражениях наряду с переменными и константами. Условимся далее называть процедуру или функцию общим именем «подпрограмма».

Подпрограммы представляют собой инструмент, с помощью которого любая программа может быть разбита на ряд в известной степени независимых друг от друга частей.

Вызов подпрограммы осуществляется простым упоминанием имени процедуры в операторе вызова процедуры или имени функции в выражении. Как известно, любое имя в программе должно быть обязательно описано перед тем как оно появится среди исполняемых операторов. Не делается исключения и в отношении подпрограмм: каждую свою процедуру и функцию программисту необходимо описать в разделе описаний.

Описание подпрограммы состоит из заголовка и тела подпрограммы.

Заголовок

Заголовок процедуры имеет вид:

PROCEDURE <имя> [(<сп. ф. п. >)] ;

Заголовок функции:

FUNCTION <имя> [(<сп.ф.п.>)] : <тип>;

Здесь <имя> - имя подпрограммы (правильный идентификатор);

<сп.ф.п.> - список формальных параметров;

<тип> - тип возвращаемого функцией результата.

Параметры

Список формальных параметров необязателен и может отсутствовать. Если же он есть, то в нем должны быть перечислены имена формальных параметров и их типы.

Все переменные из списка параметров могут использоваться в любых выражениях внутри подпрограммы. Таким способом осуществляется настройка алгоритма подпрограммы на конкретную задачу. Если параметры определяются как параметры-переменные, перед ними необходимо ставить зарезервированное слово *VAR*, а если это параметры-константы,- слово *CONST*.

C++

Функция C++ по структуре подобна программе *main*, которую вы использовали во всех предыдущих программах. Другими словами, имя функции предваряется ее типом, а за ним следует список параметров, описание которых появляется в скобках. Вы группируете операторы функций внутри левой и правой фигурных скобок, как показано ниже:

тип_возврата имя_функции (список_параметров)

```
{  
операторы;  
}
```

Если функция не возвращает значение, вам необходимо предварить имя функции типом *void*. В противном случае вы должны предварять имя функции типом возвращаемого функцией значения, например, *int*, *float*, *char* и т. д. Чтобы вернуть значение вызвавшей функции, функция использует оператор *return*. Когда программа встречает оператор *return*, выполнение функции завершается и указанное значение возвращается вызвавшей функции. Возможны ситуации, использования оператора *return* в функции, которая не возвращает значение:

```
return;
```

В этом случае функция имеет тип *void* и оператор *return* просто завершает выполнение функции.

Когда программы передают параметр в функцию, то по умолчанию C++ делает копию значения параметра и помещает эту копию во временный участок памяти, называемый стеком. Затем функция использует копию значения для выполнения своих операций. Когда функция завершается, C++ сбрасывает содержимое стека и все изменения, сделанные функцией в копиях значений параметра.

Для изменения значения параметра функция должна знать адрес памяти параметра. Чтобы сообщить функции адрес параметра, ваши программы должны использовать оператор адреса C++ (&). Внутри функции вы должны сообщить C++, что программа будет передавать параметры с помощью адреса. Для этого вы объявляете переменные-указатели, предвзяв имя каждой переменной звездочкой.

Варианты заданий

Расчет конечных сумм

В приводимых задачах необходимо составить программу расчета конечной суммы и сравнения полученного результата с контрольным значением. Число членов суммы вводится с клавиатуры с защитой от возможного неверного ввода данных (таблица 1).

Таблица 1. Конечные суммы

Вариант	Вид суммы	Контрольное значение
1	$2+4+6+8+\dots+2N$	$N(n+1)$
2	$1^2+2^2+3^2+\dots+N^2$	$\frac{N(N+1)(2N+1)}{6}$
3	$1^2+3^2+5^2+\dots+(2N-1)^2$	$\frac{N(4N^2-1)}{3}$
4	$1^3+2^3+3^3+\dots+N^3$	$\frac{N^2(N+1)^2}{4}$
5	$1^2+3^2+5^2+\dots+(2N-1)^2$	$N^2(2N^2-1)$

Ряды с факториалами

Составить программу, содержащую функцию вычисления $F(x)$ в виде бесконечного ряда с точностью 10^9 .

В основной программе организовать вычисление этого ряда для двух значений x , запрашиваемых с клавиатуры и проверку получаемых результатов путем сравнения с системной функцией $F(x)$ (табл. 2).

Таблица 2. Ряды с факториалами

Вариант	функция	ряд
6	$\text{Sin}(x)$	$x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + (-1)^N \frac{x^{2N+1}}{(2N+1)!} + \dots$
7	$\text{Exp}(x)$	$1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \dots + \frac{x^N}{N!} + \dots$
8	$\text{Cos}(x)$	$1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + (-1)^N \frac{x^{2N}}{2N!} + \dots$
9	$\text{Sh}(x)$	$x + \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + \frac{x^{2N+1}}{(2N+1)!} + \dots$
10	$\text{Ch}(x)$	$1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + \frac{x^{2N}}{2N!} + \dots$
11	$\text{Sin}(x)/x$	$1 - \frac{x^2}{3!} + \frac{x^4}{5!} + \dots + (-1)^N \frac{x^{2N}}{(2N+1)!} + \dots$

Разложение функций в ряд Фурье

Составить программу с расчетом функции $F(x)$ в виде ее разложения в ряд Фурье с точностью 10^4 (задача гармонического анализа) (табл. 3).

Таблица 2. Разложение функций в ряд Фурье

Вариант	функция	Ряд Фурье	Диапазон аргумента
---------	---------	-----------	--------------------

12	x	$2(\frac{\sin(x)}{1} - \frac{\sin(2x)}{2} + \frac{\sin(3x)}{3} - \dots)$	$-\pi < x < \pi$
13	x	$\frac{\pi}{2} - \frac{4}{\pi}(\cos(x) + \frac{\cos(3x)}{3^2} + \frac{\cos(5x)}{5^2} + \dots)$	$-\pi < x < \pi$
14	\sin(x)	$\frac{2}{\pi} - \frac{4}{\pi}(\frac{\cos(2x)}{1*3} + \frac{\cos(4x)}{3*5} + \dots)$	$-\pi < x < \pi$
15	($\pi-x$)/2	$\sum_{i=1}^{\infty} \frac{\sin(ix)}{i}$	$-\pi < x < \pi$
16	(π^2-3x^2)/12	$\sum_{i=1}^{\infty} (-1)^{i+1} \frac{\cos(ix)}{i^2}$	$-\pi < x < \pi$
17	($\pi^2x-\pi x^2$)/8	$\sum_{i=1}^{\infty} \frac{\sin(2i+1)x}{(2i+1)^3}$	$0 < x < \pi$

Расчет бесконечных сумм

В приводимых задачах необходимо составить программу расчета бесконечной суммы обратных степеней числового ряда. Суммирование проводить, пока очередной член ряда по модулю не станет меньше заданной точности ϵ . Результат сравнить с точным значением S_T . (табл. 3)

Таблица 3. Расчет бесконечных сумм

№	Вид суммы	N	Вид ряда	S_T	ϵ
18	$\sum_{i=1}^{\infty} i^{-N}$	2	$1 + \frac{1}{2^2} + \frac{1}{3^2} + \dots$	$\pi^2/6$	10^{-4}
19		4	$1 + \frac{1}{2^4} + \frac{1}{3^4} + \dots$	$\pi^4/90$	10^{-6}
20	$\sum_{i=1}^{\infty} (-1)^{i-1} i^{-N}$	2	$1 - \frac{1}{2^2} + \frac{1}{3^2} + \dots$	$\pi^2/12$	10^{-5}
21		4	$1 - \frac{1}{2^4} + \frac{1}{3^4} - \dots$	$7\pi^2/720$	10^{-7}
22	$\sum_{i=1}^{\infty} (2i+1)^{-N}$	2	$1 + \frac{1}{3^2} + \frac{1}{5^2} + \dots$	$\pi^2/8$	10^{-4}
23		4	$1 + \frac{1}{3^4} + \frac{1}{5^4} + \dots$	$\pi^4/96$	10^{-5}
24	$\sum_{i=1}^{\infty} \frac{(-1)^i}{(2i+1)^n}$	2	$1 - \frac{1}{3} + \frac{1}{5} - \dots$	$\pi/4$	10^{-4}
25		4	$1 - \frac{1}{3^3} + \frac{1}{5^3} - \dots$	$\pi^3/32$	10^{-5}

Лабораторная работа № 9

Тема. Работа с текстовыми файлами.

Цель. Научиться работать с внешними текстовыми файлами.

Задание. Написать программу для чтения из файла, обработки данных и записи результата в файл.

Теоретические сведения

Паскаль

Текстовые файлы связываются с файловыми переменными, принадлежащими стандартному типу TEXT. Текстовые файлы предназначены для хранения текстовой информации.

Связывание логического устройства с файловой переменной осуществляется процедурой *ASSIGN*

Assign(имя файловой переменной, путь к файлу)

Инициировать файл означает указать для этого файла направление передачи данных. В Турбо Паскале можно открыть файл для чтения, для записи информации, а также для чтения и записи одновременно.

Для чтения файл иницируется с помощью стандартной процедуры *RESET*:

RESET (<ф.п.>);

Здесь <ф.п.> - файловая переменная, связанная ранее процедурой *ASSIGN* с уже существующим файлом или логическим устройством-приемником информации.

Для текстовых файлов, открытых процедурой *RESET*, нельзя использовать процедуру *WRITE* или *WRITELN*.

Стандартная процедура *REWRITE* (<ф.п.>).

инициирует запись информации в файл или в логическое устройство, связанное ранее с файловой переменной <ф.п.>. Процедурой *REWRITE* нельзя инициировать запись информации в ранее существовавший дисковый файл: при выполнении этой процедуры старый файл уничтожается и никаких сообщений об этом в программу не передается. Новый файл подготавливается к приему информации и его указатель принимает значение 0.

Стандартная процедура *APPEND* (<ф.п.>)

инициирует запись в ранее существовавший текстовый файл для его расширения, при этом указатель файла устанавливается в его конец. Процедура *APPEND* применима только к текстовым файлам, т.е. их файловая переменная должна иметь тип *TEXT*

Процедура *Close*.

Закрывает файл, однако связь файловой переменной с именем файла, установленная ранее процедурой *ASSIGN*, сохраняется. Формат обращения:

CLOSE (<ф.п.>)

Функция *EOF* (<ф. п. >) : *BOOLEAN*.

Логическая функция, тестирующая конец файла. Возвращает *TRUE*, если файловый указатель стоит в конце файла. При записи это означает, что очередной компонент будет добавлен в конец файла, при чтении - что файл исчерпан.

Процедура *READ*.

Обеспечивает ввод символов, строк и чисел. Формат обращения:

READ (<ф.п.>, <сп.ввода>) или *READ* (<сп.ввода>)

Здесь <сп.ввода> - список ввода: последовательность из одной или более переменных типа *CHAR*, *STRING*, а также любого целого или вещественного типа.

При вводе переменных типа *CHAR* выполняется чтение одного символа из файла и присваивание считанного значения переменной.

Процедура *READLN*.

Обеспечивает ввод символов, строк и чисел. Эта процедура идентична процедуре *READ* за исключением того, что после считывания последней переменной оставшаяся часть строки до маркера *EOLN* пропускается, поэтому следующее обращение к *READLN* или *READ* начинается с первого символа новой строки.

Процедура *WRITE*.

Обеспечивает вывод информации в текстовый файл или передачу ее на логическое устройство. Формат обращения:

WRITE (<ф.п.>, <сп.вывода>) или *WRITE* (<сп.вывода>)

Здесь <сп.вывода> - список вывода: последовательность из одного или более выражений типа *CHAR*, *STRING*, *BOOLEAN*, а также любого целого или вещественного типа.

Файловая переменная <ф.п.>, если она указана, должна быть предварительно описана как переменная типа *TEXT* и связана с именем файла или логическим устройством процедурой *ASSIGN*.

Процедура *WRITELN*. Эта процедура полностью идентична процедуре *WRITE* за исключением того, что после вывода происходит переход на новую строку.
Логическая функция *EOLN*. Возвращает *TRUE*, если во входном текстовом файле достигнут маркер конца строки. Формат обращения: *EOLN*<*ф.п.*>

C++

Основные концепции при работе с файлами в C++:

- Используя выходной файловый поток, вы можете писать информацию в файл с помощью оператора вставки (<<).
- Используя входной файловый поток, вы можете читать хранимую в файле информацию с помощью оператора извлечения (>>).
- Для открытия и закрытия файла вы используете методы файловых классов.

Переменные потока ввода из файла имеют тип *ifstream*. переменные потока вывода в файл имеют тип *ofstream*. оба типа описаны в библиотеке с заголовочным файлом *fstream.h*.

Файл открывается функцией:

имя файловой переменной.open(имя файла)

Файл закрывается функцией:

имя файловой переменной.close()

проверка ошибок при выполнении файловых операций

Чтобы помочь вашим программам следить за ошибками, вы можете использовать функцию *fail* файлового объекта. Если в процессе файловой операции ошибок не было, функция возвратит ложь (0). Однако, если встретилась ошибка, функция *fail* возвратит истину.

Чтобы определить конец файла, ваши программы могут использовать функцию *eof* потокового объекта. Эта функция возвращает значение 0, если конец файла еще не встретился, и 1, если встретился конец файла. Используя цикл *while*, ваши программы могут непрерывно читать содержимое файла, пока не найдут конец файла, как показано ниже:

```
while (! input_file.eof())  
{  
    // Операторы  
}
```

Кроме уже описанных процедур файл можно открывать другим способом, а имен сразу после объявления файловой переменной. Однако в этом случае необходимо указывать режимы открытия и тип переменной будет: *fstream*.

fstream имя переменной («имя файла», режим);

Значения режимов открытия.

Режим открытия	Назначение
<i>ios::app</i>	Открывает файл в режиме добавления, располагая файловый указатель в конце файла.
<i>ios::ate</i>	Располагает файловый указатель в конце файла.
<i>ios::in</i>	Указывает открыть файл для ввода.
<i>ios::nocreate</i>	Если указанный файл не существует, не создавать файл и вернуть ошибку.
<i>ios::noreplace</i>	Если файл существует, операция открытия должна быть прервана и должна вернуть ошибку.
<i>ios::out</i>	Указывает открыть файл для вывода.
<i>ios::trunc</i>	Сбрасывает (перезаписывает) содержим, з существующего файла.

Следующая операция открытия файла открывает файл для вывода, используя режим *ios::noreplace*, чтобы предотвратить перезапись существующего файла:

ifstream output_file("Filename.EXT", ios::out | ios::noreplace);

Варианты заданий

1. Исключить из массива $A_1..A_N$ максимальный элемент.
2. Исключить из массива $A_1..A_N$ первый отрицательный элемент.
3. Даны действительные числа $x_1, \dots, x_{11}, y_1, \dots, y_{11}$. Получить $x*s_1, \dots, x*s_{11}, y*s_1, \dots, y*s_{11}$, преобразовав члены x_i, y_i по правилу: если они оба отрицательны, то каждый из них увеличить на 0.5; если отрицательно только одно число, то отрицательное число заменить его квадратом; если оба числа неотрицательны, то каждое из них заменить на среднее арифметическое исходных значений.
4. Даны целые числа a_1, \dots, a_n . Определить количество целых чисел, входящих в последовательность a_1, \dots, a_n по одному разу.
5. Даны целые числа a_1, \dots, a_n . Из модулей членов данной последовательности выбрать наибольший. Получить новую последовательность из n целых чисел, заменяя a_i нулем, если $|a_i|$ не совпадает с выбранным значением, и заменяя a_i единицей, если совпадает.
6. Перенести в хвост одномерного массива максимальный элемент.
7. Даны действительные числа a_1, \dots, a_n . Требуется найти B равное среднему арифметическому чисел a_1, \dots, a_n .
8. Даны 2 последовательности действительных чисел x_1, \dots, x_n и y_1, \dots, y_n . Выяснить, верно ли, что среди точек (x_i, y_i) есть хотя бы одна, принадлежащая квадрату, стороны которого параллельны координатным осям, центр совпадает с началом координат а длина стороны равна 2. Вывести координаты точек, удовлетворяющих этому условию.
9. Даны 2 последовательности действительных чисел $x_1, \dots, x_{10}, e_1, \dots, e_{10}$. Выяснить верно ли, что все точки (x_i, y_i) принадлежат кругу радиуса 2 с центром в точке $(1, 1)$.
УКАЗАНИЕ: воспользуйтесь формулой $R^2 = x^2 + y^2$ для круга с центром в точке $(0, 0)$.
10. Дан массив целых чисел a_1, \dots, a_n . Выяснить, какая из трех ситуаций имеет место: все числа a_1, \dots, a_n равны нулю, в последовательности a_1, \dots, a_n первое ненулевое число - положительное, первое ненулевое число - отрицательное.
11. Перенести в хвост одномерного массива первый отрицательный элемент.
12. Подсчитать в одномерном массиве максимальное количество подряд идущих нулей.
13. Перенести в начало одномерного массива второй нулевой элемент.
14. Подсчитать в одномерном массиве максимальное количество серий подряд идущих единиц.
15. Исключить из массива $A_1..A_N$ первую серию отрицательных элементов.
16. Перенести в хвост одномерного массива все отрицательные элементы.
17. Перенести в начало одномерного массива все нечетные элементы.
18. Исключить из массива $A_1..A_N$ первый четный элемент, следующий за максимальным.
19. Исключить из массива $A_1..A_N$ первое число, большее 3.
20. Дан массив действительных чисел a_1, \dots, a_n . Найти максимальный элемент среди отрицательных элементов и поменять его местами с минимальным положительным.
21. Дан массив слов a_1, \dots, a_n . Найти предпоследнее слово с максимальной длиной и напечатать его значение и индекс.
22. Дан массив целых чисел a_1, \dots, a_n . Найти сумму двух самых больших по значению элементов массива и напечатать ее значение.
23. Дан массив действительных чисел a_1, \dots, a_n . Найти первый отрицательный элемент массива и напечатать его значение и индекс.
24. Даны действительные числа a_1, \dots, a_{16} . Получить новый массив по правилу $(a_1+a_{16}, a_2+a_{15}, \dots, a_8+a_9)$. Найти минимальный элемент полученного массива.
25. Даны целые числа a_1, \dots, a_{16} . Получить новый массив по правилу $(a_1*a_9, a_2*a_{10}, \dots, a_8*a_{16})$. Найти минимальный элемент полученного массива.

Лабораторная работа № 9

Тема. Работа со структурированными типами данных.

Цель. Научиться работать с записями.

Задание. Разработать программу со следующими функциями:

- ввод данных структуры;
- сохранения данных в файл;
- чтение данных из файла;
- вывод на экран таблицы с данными;
- выполнение указанного в варианте запроса, по данным веденным пользователем.

Теоретические сведения

Паскаль

Запись - это структура данных, состоящая из фиксированного числа компонентов, называемых полями записи. В отличие от массива, компоненты (поля) записи могут быть различного типа. Чтобы можно было ссылаться на тот или иной компонент записи, поля именуются.

Структура объявления типа записи такова:

<имя типа> = RECORD <сп.полей> END

Здесь *<имя типа>* - правильный идентификатор;

<сп.полей> - список полей; представляет собой последовательность разделов записи, между которыми ставится точка с запятой.

Каждый раздел записи состоит из одного или нескольких идентификаторов полей, отделяемых друг от друга запятыми. За идентификатором (идентификаторами) ставится двоеточие и описание типа поля (полей), например:

type

BirthDay = record

day,month : Byte;

year : Word

end;

var

a,b : Birthday;

.....

В этом примере тип BIRTHDAY (день рождения) есть запись с полями DAY, MONTH и YEAR (день, месяц и год); переменные A и B содержат записи типа BIRTHDAY.

Как и в массиве, значения переменных типа записи можно присваивать другим переменным того же типа, например

a := b;

К каждому из компонентов записи можно получить доступ, если использовать составное имя, т.е. указать имя переменной, затем точку и имя поля:

a.day := 27;

b.year := 1939;

Для вложенных полей приходится продолжать уточнения

Чтобы упростить доступ к полям записи, используется оператор присоединения WITH:

WITH <переменная> DO <оператор>

<переменная> - имя переменной типа запись, за которым, возможно, следует список вложенных полей; *<оператор>* - любой оператор Турбо Паскаля.

Например:

with c.bd do month := 9;

Это эквивалентно

with c do with bd do month := 9;

или

with c, bd do month := 9;

или

c.bd.month := 9;

Турбо Паскаль разрешает использовать записи с так называемыми вариантными полями, например:

type

Forma = record

Name: String;

case Byte of

0: (Birthplace: String [40]);

1: (Country : String [20];

EntryPort : String [20];

EntryDate : 1..31;

ExitDate : 1..31)

end;

В этом примере тип **FORMA** определяет запись с одним фиксированным полем **NAME** и вариантной частью, которая задается предложением **CASE... OF**. Вариантная часть состоит из нескольких вариантов (в примере - из двух вариантов: 0 и 1). Каждый вариант определяется константой выбора, за которой следует двоеточие и список полей, заключенный в круглые скобки. В любой записи может быть только одна вариантная часть, и, если она есть, она должна располагаться за всеми фиксированными полями.

Замечательной особенностью вариантной части является то обстоятельство, что все заданные в ней варианты «накладываются» друг на друга, т.е. каждому из них выделяется одна и та же область памяти.

C++

Структура определяет шаблон, с помощью которого программа может позднее объявить одну или несколько переменных. Другими словами, программа сначала определяет структуру, а затем объявляет переменные типа этой структуры. Для определения структуры используется ключевое слово *struct*, за которым обычно следует имя и левая фигурная скобка. Следом за открывающей фигурной скобкой указывается тип и имя одного или нескольких *элементов*. За последним элементом вы размещаете правую закрывающую фигурную скобку.

Struct имя структуры

{

поля структуры

}

Варианты заданий

1. Каталог книг:

- Название.
- Автор
- Количество страниц.
- Год издания.

Запрос. Вывести на экран все книги данного автора.

2. Каталог газет:

- Название газеты.
- Номер.
- Дата выхода.
- Количество страниц.

Запрос. Вывести на экран все газеты вышедшие в определенном месяце.

3. Каталог газетных статей:

- Автор.
- Название.
- Название газеты или журнала.
- Номер газеты или журнала.

- Номер страницы.

Запрос. Вывести на экран все статьи данного автора.

4. Перечень факультетов:

- Название факультета.
- ФИО декана.
- Телефон.
- Адрес.

Запрос. Вывести на экран декана данного факультета.

5. Перечень кафедр:

- Название кафедры.
- ФИО зав.каф.
- Количество преподавателей.
- Адрес.

Запрос. Вывести на экран кафедры, где количество преподавателей превышает заданное.

6. Перечень групп:

- Наименование группы.
- Курс.
- Староста.
- Количество студентов.

Запрос. Вывести на экран группы, где количество студентов меньше заданного

7. Перечень студентов:

- ФИО студента.
- Дата рождения.
- Адрес.
- Телефон

Запрос. Вывести на экран студентов имеющих одинаковую дату рождения.

8. Рейтинг успеваемости студентов:

- ФИО студента.
- Группа.
- Средний бал.
- Размер стипендии.

Запрос. Вывести на экран студентов, средний бал которых превышает заданный.

9. Перечень основных дисциплин:

- Название дисциплины.
- Кафедра на которой читается дисциплина.
- ФИО преподавателя, читающего лекции.
- ФИО преподавателя, ведущего лабораторные занятия.

Запрос. Вывести на экран дисциплины, которые читаются преподавателями заданной кафедры.

10. Основные характеристики дисциплины:

- Название.
- Количество лекционных часов.
- Количество лабораторных.
- Количество практических.
- Количество ИРК.
- Вид семестрового контроля(экзамен, зачет).

Запрос. Вывести на экран дисциплины отсортированные по виду семестрового контроля (сначала. Те дисциплины по которым экзамен, а потом те по которым зачет), также указывать наличие курсового или индивидуального.

11.Список дисциплин кафедры:

- Название.
- ФИО преподавателя.
- Семестр в котором читается дисциплина.
- Группа.

Запрос. Вывести на экран дисциплины отсортированные по преподавателям.

12. Расписание преподавателя:

- Дата.
- День недели.
- Предмет.
- Группа.
- Аудитория.

Запрос. Вывести на экран все пары преподавателя с заданной группой.

13.Список читателей:

- Номер читательского.
- ФИО читателя.
- Адрес.
- Телефон.

Запрос. Вывести на экран всех читателей, которые проживают на заданной улице.

14.Список выданных книг:

- Код книги.
- Номер читательского.
- Дата выдачи.
- Срок выдачи.

Запрос. Вывести на экран книги, которые читатели не сдали вовремя на заданное число.

15.Список товаров:

- Наименования товара.
- Единица измерения.
- Количество на складе.
- Цена за единицу.

Запрос. Рассчитать общую стоимость каждого товара на складе.

16. Список покупателей.

- Наименование.
- Адрес.
- Телефон.
- Код.

Запрос. Вывести на экран всех покупателей, которые подключены к одной АТС.

17.Список продаж:

- Наименование товара.
- Наименование покупателя.
- Дата продажи.
- Количество.
- Стоимость.

Запрос. Вывести на экран суммарную стоимость покупок каждого покупателя.

18. Список продаж за день:

- Дата.
- Количество проданных товаров.
- Сумма оплаченных продаж.
- Сумма продаж в кредит.

Запрос. Вывести на экран суммы продаж по каждому дню.

19. Справочник районов города:

- Наименование района.
- ФИО главы администрации.
- Общая площадь.
- Количество административных единиц.

Запрос. Вывести на экран районы, площадь которых превышает заданную.

20. Справочник улиц города:

- Наименование улицы.
- Длина.
- История.
- Район.

Запрос. Вывести на экран улицы отсортированные по районам.

21. Справочник городов:

- Название города.
- ФИО мэра.
- Площадь.
- Среднее количество населения.

Запрос. Вывести на экран города, среднее количество населения которых превышает заданное.

22. Справочник стран:

- Наименование.
- Столица.
- Вид правления.
- Глава государства.

Запрос. Вывести на экран страны с заданным видом правления.

23. Каталог журналов:

- Название.
- Вид.
- Номер.
- Дата выхода.
- Количество страниц.

Запрос. Вывести на экран журналы, количество страниц которых превышает заданное.

24. Справочник сотрудников предприятия:

- Табельный номер.
- ФИО.
- Пол.
- Дата рождения.
- Должность.

Запрос. Вывести на экран количество сотрудников по каждой из должностей.

25. Справочник пропусков студентов.

- ФИО студента.
- Группа.
- Количество пропусков.

- Количество неаттестаций.

Запрос. Вывести на экран студентов, количество пропусков которых превышает заданное.