Министерство образования и науки Р Ф

Томский государственный университет систем управления

и радиоэлектроники

Кафедра радиоэлектроники и защиты информации (РЗИ)

**Контрольная работа № 1**

по дисциплине «Информатика-2»

учебное пособие Гураков А.В., Мещеряков П.С. «Информатика. Часть 2»

**Вариант №**

Выполнил студент

направления подготовки

Томск — 2019

**Содержания**

[Задание 1 3](#_Toc516121673)

[Решение 3](#_Toc516121674)

[Алгоритм 3](#_Toc516121675)

[Описание структуры программы 4](#_Toc516121676)

[Код программы 4](#_Toc516121677)

[Результаты работы программы 4](#_Toc516121678)

[Задание 2 5](#_Toc516121679)

[Решение 5](#_Toc516121680)

[Алгоритм 5](#_Toc516121681)

[Описание структуры программы 6](#_Toc516121682)

[Код программы 6](#_Toc516121683)

[Результаты работы программы 7](#_Toc516121684)

[Задание 3 8](#_Toc516121685)

[Решение 8](#_Toc516121686)

[Описание структуры программы 8](#_Toc516121687)

[Алгоритм 9](#_Toc516121688)

[Код программы 11](#_Toc516121689)

[Результаты работы программы 11](#_Toc516121690)

**Задание 1**

Дано вещественное число а. Для функции y = f(x), график которой приведен ниже, вычислить f(а).



**Решение:**

График функции можно задать формулой:

$$f=\left\{\begin{array}{c}-3, a<-1.25\\tg\left(a\right), -1.25\leq a\leq 1.25 \\3, a>1.25.\end{array}\right.$$

**Алгоритм**:

Начало

Ввод a

Конец

a<-1.25

F=-3

a<1.25

F=sin(a)/cos(a)

F=3

Ввод F

**Описание структуры программы:**

Для решения задачи достаточно двух переменных: *А* и F. В переменную А вводится значение аргумента, в переменную F записывается результат вычисления f(a). Обе переменные вещественного типа.

**Код программы:**

program prog1;

var a,f:real;

begin

write('Введите значения числа а ');

readln(a);

if a<-1.25 then f:=-3

else

if a<=1.25 then f:=sin(a)/cos(a)

else

f:=3;

writeln('f(',a:2:2,')=',f:2:3);

end.

**Результаты работы программы:**

Для тестирования программы использовалось значение a=-2, которое лежит на промежутке a$<-1.25 $, значение числа a=0, которое лежит на промежутке $-1.25\leq a\leq 1.25 $и a=2, которое лежит на промежутке a>$1.25$.







**Задание 2**

Вычислить сумму натуральных нечетных чисел, не превышающих N. **Решение**:

Нужно найти сумму S=1+3+5+…+N.

При N=1, то S1=1

При N=2, то S2=1+3= S1+3

При N=3, то S3=1+3+5= S2+5

Si= Si-1+i, i=1... N, i – нечетное.

Начальное значение суммы равно 0.

**Алгоритм**:

Начало

Ввод N

i<=N

S=S+i

i:=i+2

S

Конец

да

нет

Ввод N

Проверяется условие i<=N

Последовательность закончилась

i=1

S=0

Начальное значение i

Начальное значение S

Увеличиваем S на i раз

Увеличиваем i на 2

Вывод результата

**Описание структуры программы:**

Необходимо минимум две переменных для реализации данного алгоритма. Так как нужно найти сумму первых N нечетных натуральных чисел, то тип данных - целочисленный тип.

Так как известно количество чисел в сумме, но нужны только нечетные числа, поэтому счетчик должен изменятся на два, то в программе лучше всего использовать цикл While или Repeat. Начальное значения счетчика 1, суммы 0.

Для цикла For необходимо добавить дополнительную проверку счетчика на нечетность.

**Код программы:**

program prog2;

var N,s:word;i:byte;

begin

write('Введите значения N=');

readln(n);

writeln('Цикл For');

S:=0;

for i:=1 to n do

if i mod 2 =1 then S:=S+i;

writeln('S=',S);

writeln('Цикл While');

S:=0;

i:=1;

while i<=n do

begin

 S:=S+i;

 i:=i+2;

end;

writeln('S=',S);

writeln('Цикл repeat');

S:=0;

i:=1;

repeat

 S:=S+i;

 i:=i+2;

until i>N;

writeln('S=',S);

end.

**Результаты работы программы:**

Для тестирования программы использовалось значение n=8, для которого S=1+3+5+7=16



**Задание 3**.

Составить программу «сжатия» исходной строки символов: каждая подстрока, состоящая из нескольких вхождений одного и того же символа, должна быть заменена на текст «x (k)», где x — символ, а k — строка, являющаяся записью числа вхождений символа в исходную строку.

**Решение**:

Для сжатия строки, будем использовать дополнительную строку, в которую будем записывать каждый символ начальной строки и сколько раз он в ней повторяется.

Для того, чтобы узнать количество повторений подряд каждой буквы начальной строки, считаем сколько раз x[i]=x[i+1].

В программе нужно использовать два цикла. Первый цикл позволяет просмотреть все символы строки, а второй посчитать сколько раз подряд повторяется символ строки.

В конце программы поменяем начальную строку на дополнительную.

**Описание структуры программы:**

Необходимо минимум четыре переменных для реализации данного алгоритма. Х – данная строка, У – дополнительная строка, К – количество повторений подряд каждого символа строки, i – счетчик, который пробегает все символы строки.

По сколько мы не знаем длину вводимой строки, и она будет изменятся каждый раз при запуске программы, то в программе лучше всего использовать цикл While или Repeat

**Алгоритм**:

Начало

Конец

i<=length(x)

Ввод Х

k=1

(i<=length(x)-1) **and**(x[i]=x[i+1])

y=y+x[i]+'('+inttostr(k)+')';

i=1

k=k+1

i=i+1

x=y

Ввод Х

i=i+1

да

да

нет

нет

Просматриваем все символы строки

Предполагаем, что каждый символ, который есть в строке встречается хотя бы один раз.

Подсчитываем сколько раз подряд входит каждый символ в строку.

В дополнительной строке записываем символ и сколько раз он повторяется подряд в строке.

Изменяем начальную строку и выводим ее

**Код программы:**

program prog3;

var x,y:string;i,k:byte;

begin

writeln('Введите строку ');

readln(x);

i:=1;

y:='';

while i<=length(x) do

 begin

 k:=1;

 while (i<=length(x)-1)and(x[i]=x[i+1]) do

 begin

 i:=i+1;

 k:=k+1;

 end;

 y:=y+x[i]+'('+inttostr(k)+')';

 i:=i+1;

 end;

 x:=y;

 writeln(x);

 end.

**Результаты работы программы:**

