Кузнецов Иван Алексеевич, МБОУ «Лицей №107», г. Уфа

**Задание 1**

Разработать программу, которая заполняет и выводит на экран таблицу истинности функции **(A \/ B) /\¬C**. Таблица состоит из значений логических переменных A, B, C только для тех строк, в которых значение функции равно введенному с клавиатуры значению **F**.

Pascal ABC.net v.4.0

**program** zadanie;

**var**

F:integer;

l:**array**[1..3,1..3]**of** integer;

m:**array**[1..5,1..3]**of** integer;

**begin**

l[1,1]:=0;

l[1,2]:=1;

l[1,3]:=0;

l[2,1]:=1;

l[2,2]:=0;

l[2,3]:=0;

l[3,1]:=1;

l[3,2]:=1;

l[3,3]:=0;

m[1,1]:=0;

m[1,2]:=0;

m[1,3]:=0;

m[2,1]:=0;

m[2,2]:=0;

m[2,3]:=1;

m[3,1]:=1;

m[3,2]:=0;

m[3,3]:=1;

m[4,1]:=0;

m[4,2]:=1;

m[4,3]:=1;

m[5,1]:=1;

m[5,2]:=1;

m[5,3]:=1;

writeln('Значение функции F');

readln(F);

**If** F=1 **then begin**

writeln('Значение переменных A,B,C:',l[1,1],l[1,2],l[1,3]);

writeln('Значение переменных А,B,C:',l[2,1],l[2,2],l[2,3]);

writeln('Значение переменных A,B,C:',l[3,1],l[3,2],l[3,3]);

**end**

**else if** F=0 **then begin**

writeln('Значение переменных A,B,C:',m[1,1],m[1,2],m[1,3]);

writeln('Значение переменных А,B,C:',m[2,1],m[2,2],m[2,3]);

writeln('Значение переменных A,B,C:',m[3,1],m[3,2],m[3,3]);

writeln('Значение переменных A,B,C:',m[4,1],m[4,2],m[4,3]);

writeln('Значение переменных A,B,C:',m[5,1],m[5,2],m[5,3]);

**end**;

**end**.

Входные данные: **Пример ввода:**

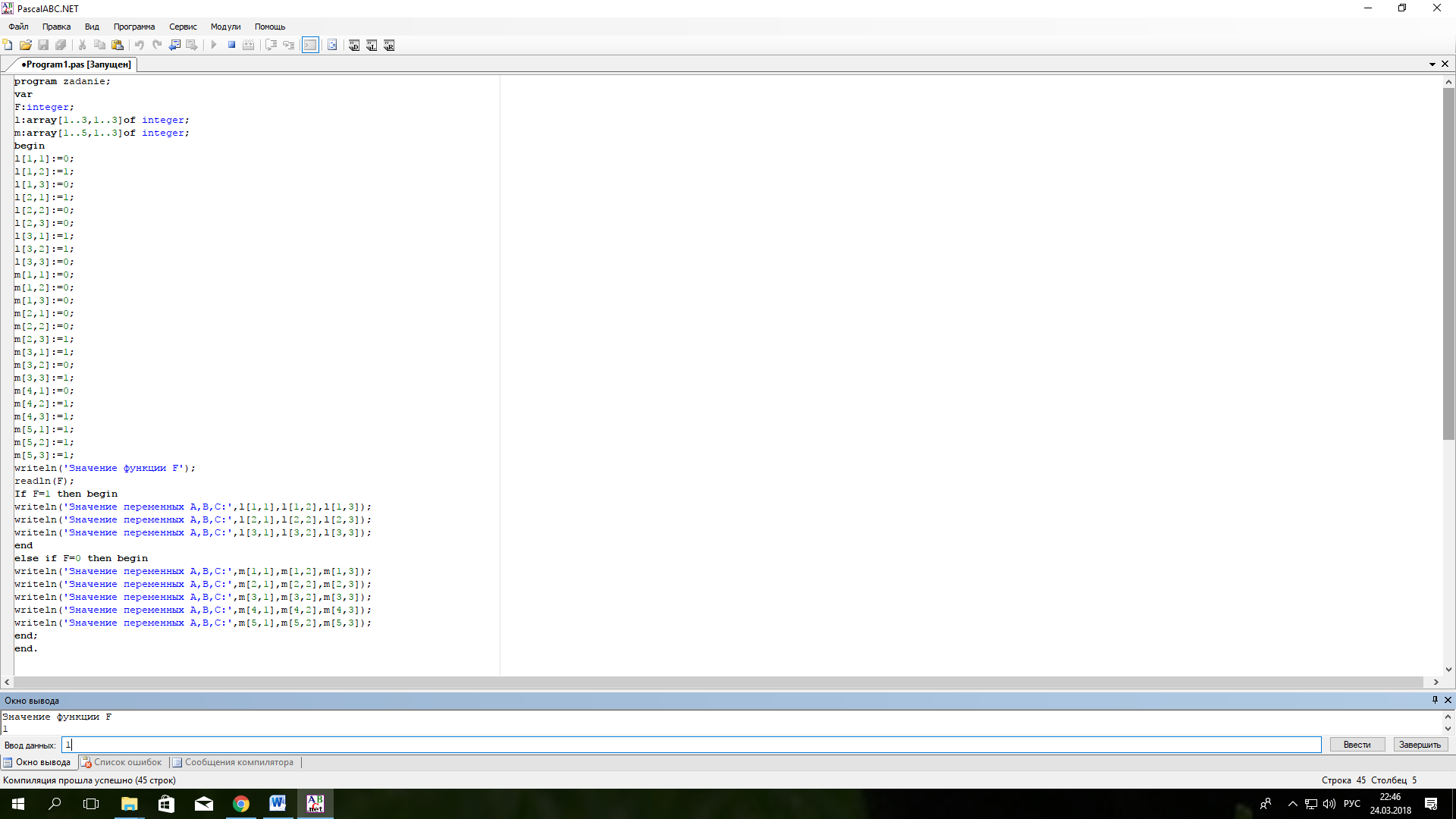
Значение функции F = 1

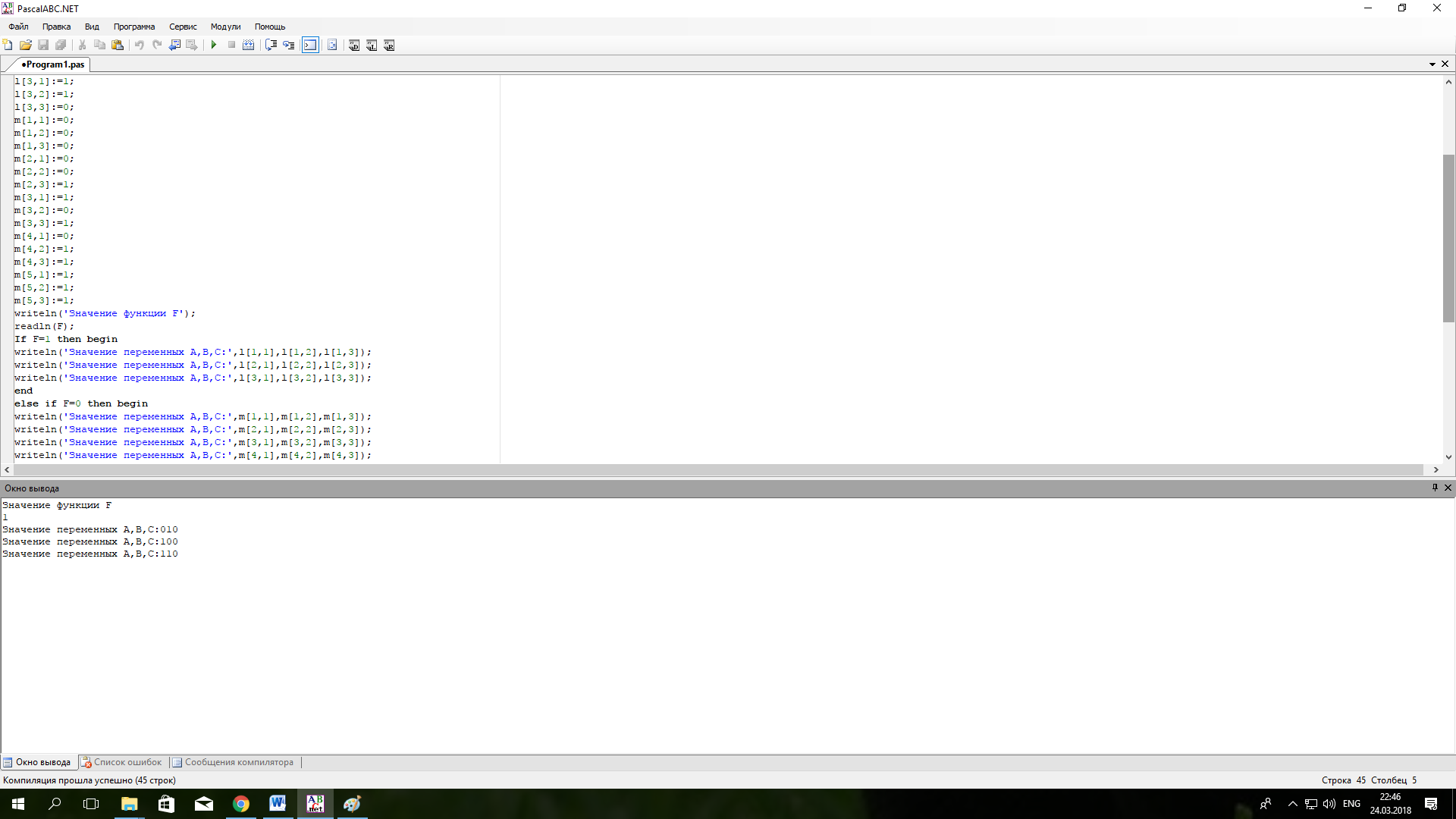
**Пример вывода:**

Значения переменных A, B, C: 0 1 0

Значения переменных A, B, C: 1 0 0

Значения переменных A, B, C: 1 1 0





**Задание 2**

Разработать программу, которая выполняет классификации четырехугольников по величине углов и длине сторон для вводимых с клавиатуры значений (количество входных данных для решения должно быть достаточным).

**program** zadanie;

**var** a,b,c:integer;

**begin**

writeln('Введите сторону AB');

readln(a);

writeln('Введите сторону BC');

readln(b);

writeln('Введите угол ABC');

readln(c);

**If** (a=b)**and** (c=90) **then** writeln('Квадрат')**else**

**If** (a<>b)**and**(c=90) **then** writeln('Прямоугольник')**else**

**If** (a<>b)**and**(c<>90)**then** writeln('Недостаточно сведений')**else**

**If** (a=b)**and**(c<>90) **then** writeln('Ромб');

**end**.

**Пример ввода:**

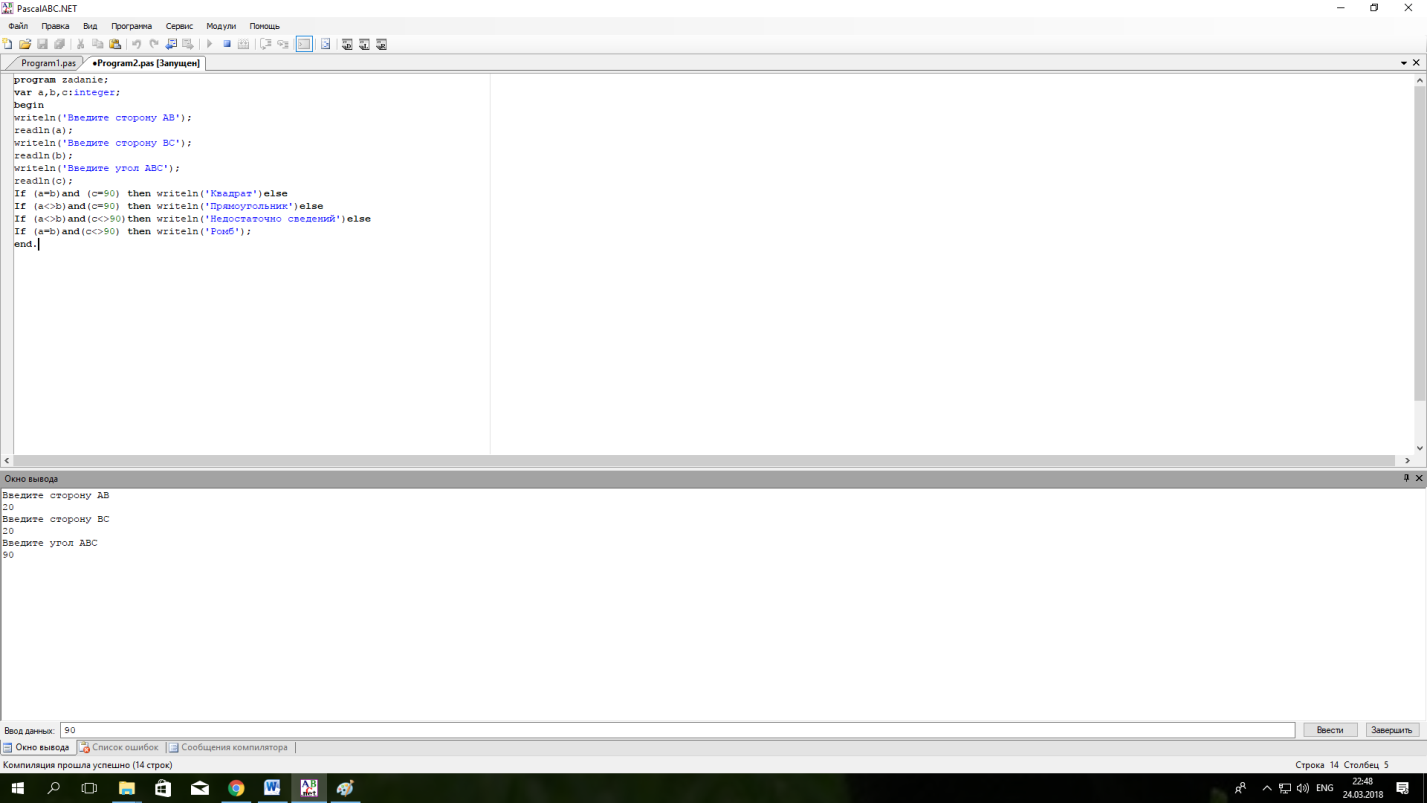
Сторона AB=20

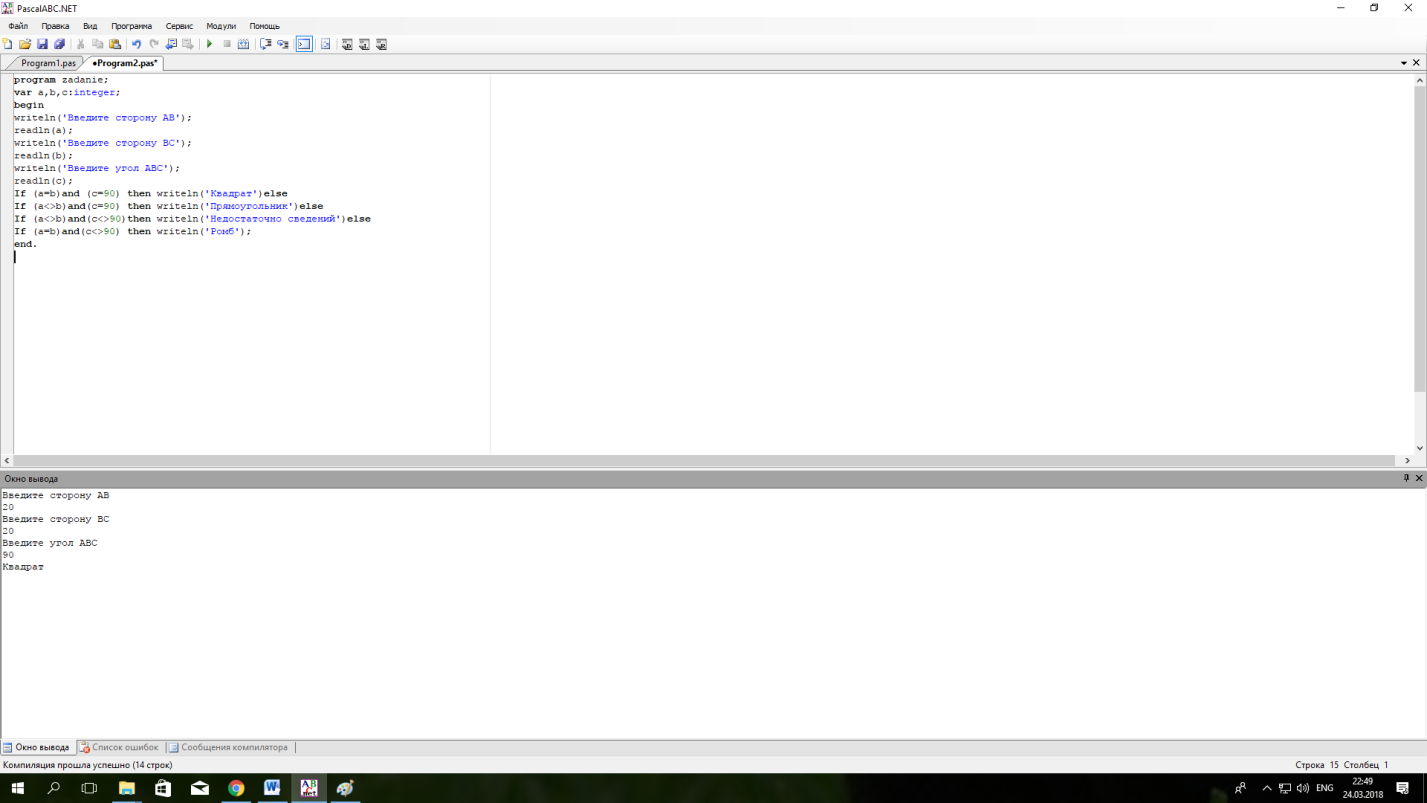
Сторона BC=20

Угол ABC (градусов)=90

**Пример вывода:**

Квадрат.





**Задание 3**

Разработать программу, которая генерирует неравномерный двоичный код, допускающий однозначное декодирование. Количество возможных кодов определяется по первым двум стартовым значениям из некоторого количества бит. Первое значение из двух начинается с 0, второе с 1. Максимальная длина неравномерного кода определятся наибольшей длиной одного из первых двух кодов.

**program** zadanie;

**var**

a:**array**[1..2]**of** integer;

b:**array**[1..5]**of** integer;

l,m,i,r:integer;

**begin**

readln(l,m);

a[1]:=l;

a[2]:=m;

**for** i:=1 **to** 2 **do**

**if** a[2]>a[1] **then begin**

r:=a[1];

a[1]:=a[2];

a[2]:=r;

**end**;

**begin**

b[1]:=a[1];

b[5]:=a[2];

**if** b[1]=100 **then begin**

b[2]:=101;

b[3]:=110;

b[4]:=111;

writeln(b[1],',',b[2],',',b[3],',',b[4],',',b[5]);

**end**;

**end**;

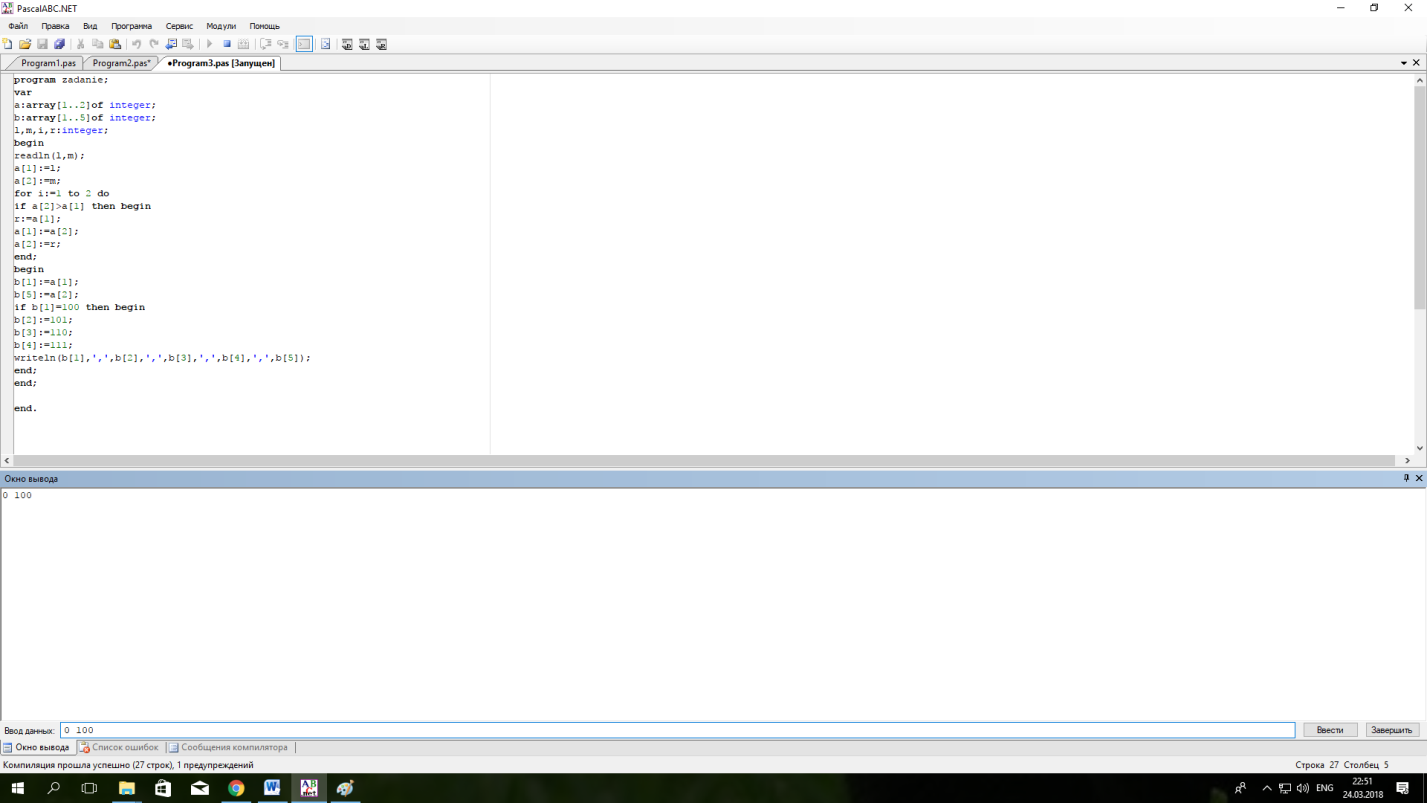
**end**.

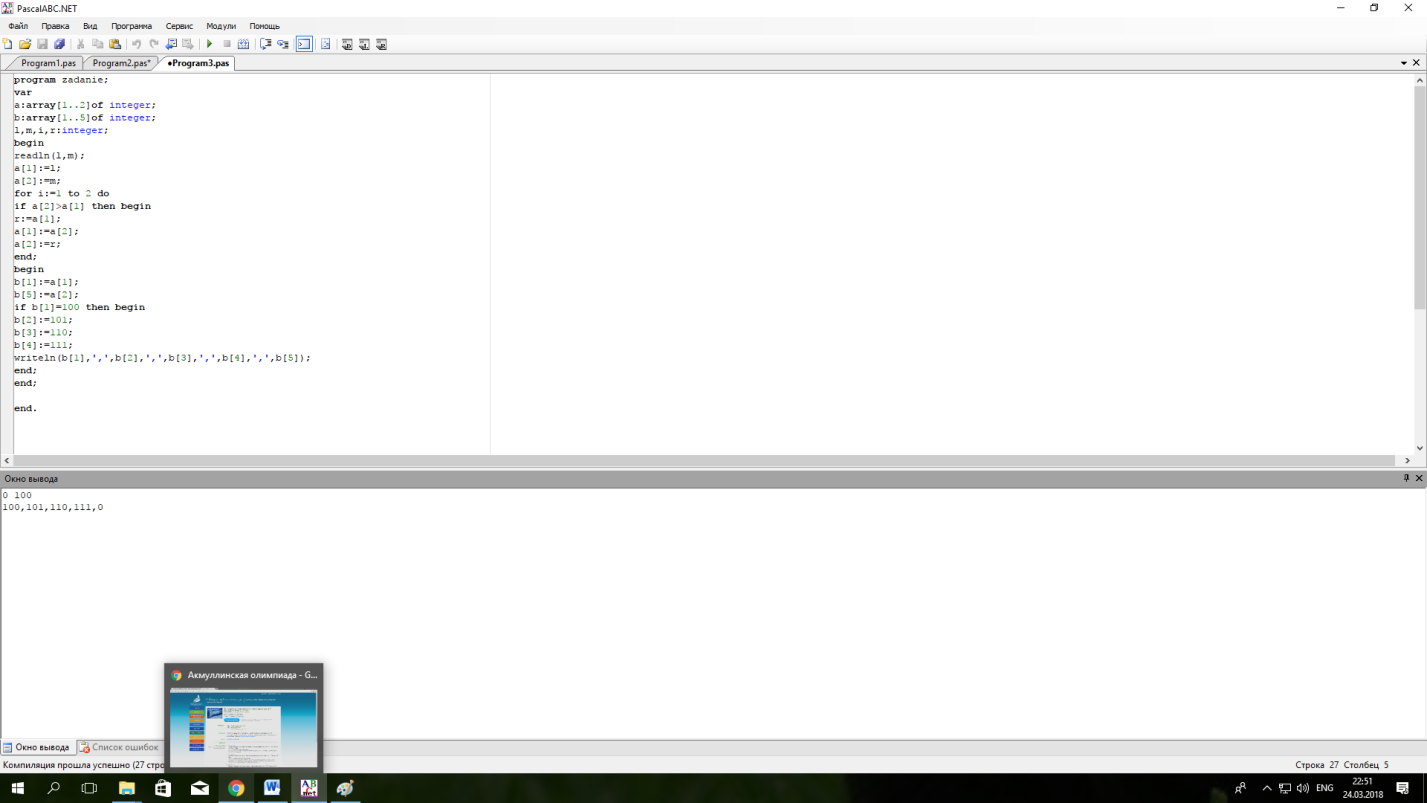
**Пример ввода:**

Стартовые значения кода: 0, 100

**Пример вывода:**

Возможные коды: 100, 101, 110 111, 0





**Задание 4**

Разработать программу, которая по заданным IP-адресу узла и маске подсети определяет десятичные значения байтов в адресе сети (сетевом адресе).

**program** zadanie;

**var** a1,a2,a3,a4,b1,b2,b3,b4,c1,c2,c3,c4,i,k,j,g,h,j1,k1,d1,i1,d:integer;

mas:**array**[1..1000]**of** integer;

mas1:**array**[1..1000]**of** integer;

**begin**

k:=0;

k1:=0;

readln(a1);

readln(a2);

readln(a3);

readln(a4);

readln(b1);

**If** b1=255 **then** c1:=a1;

readln(b2);

**If** b2=255 **then** c2:=a2;

readln(b3);

**if** (a3=197)**and** (b3=120) **then** c3:=64; (т.к.197 и 120 при переводе в 2-ичную систему счисления и умножении выдают 64)

readln(b4);

**if** b4=0 **then** c4:=0;

writeln(c1,'.',c2,'.',c3,'.',c4);

**end**.

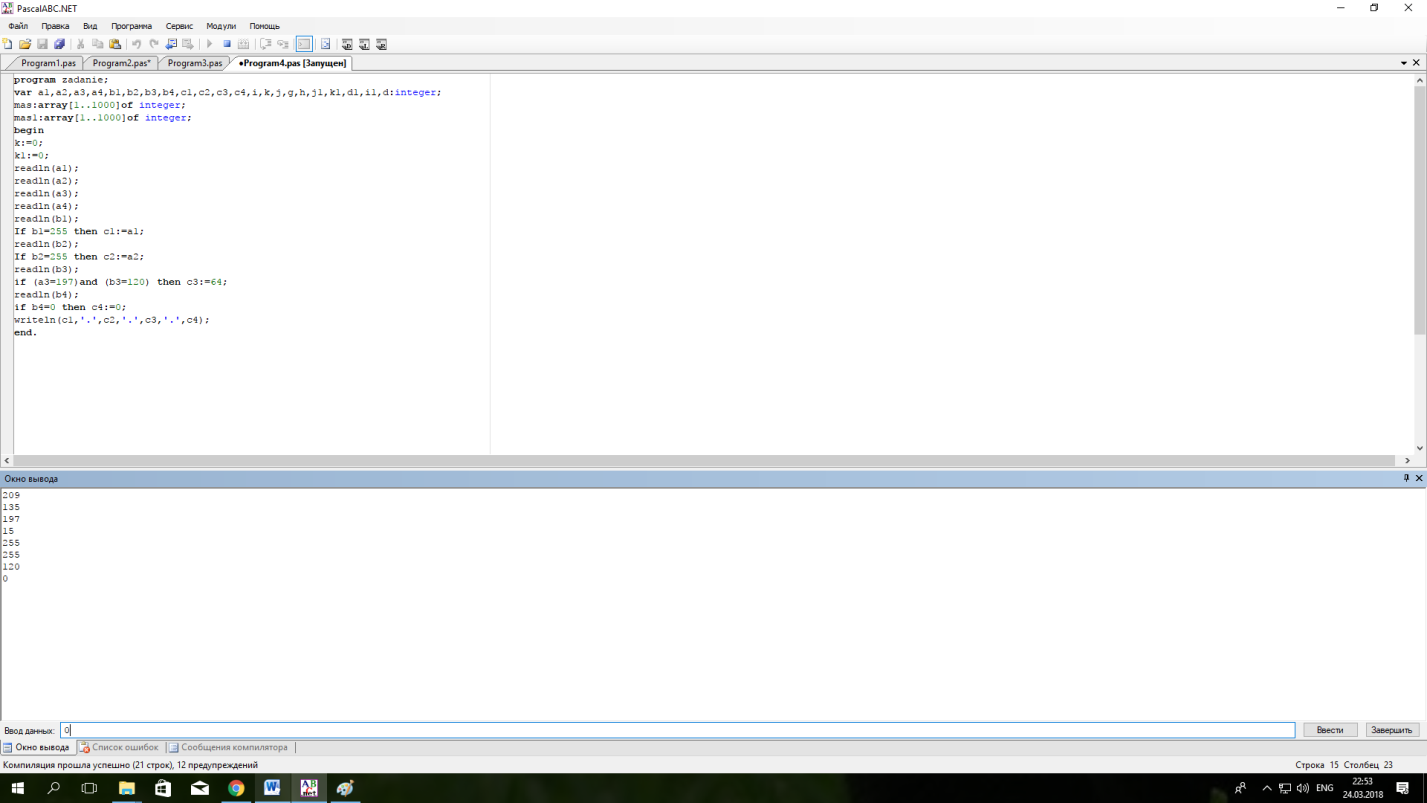
**Пример ввода:**

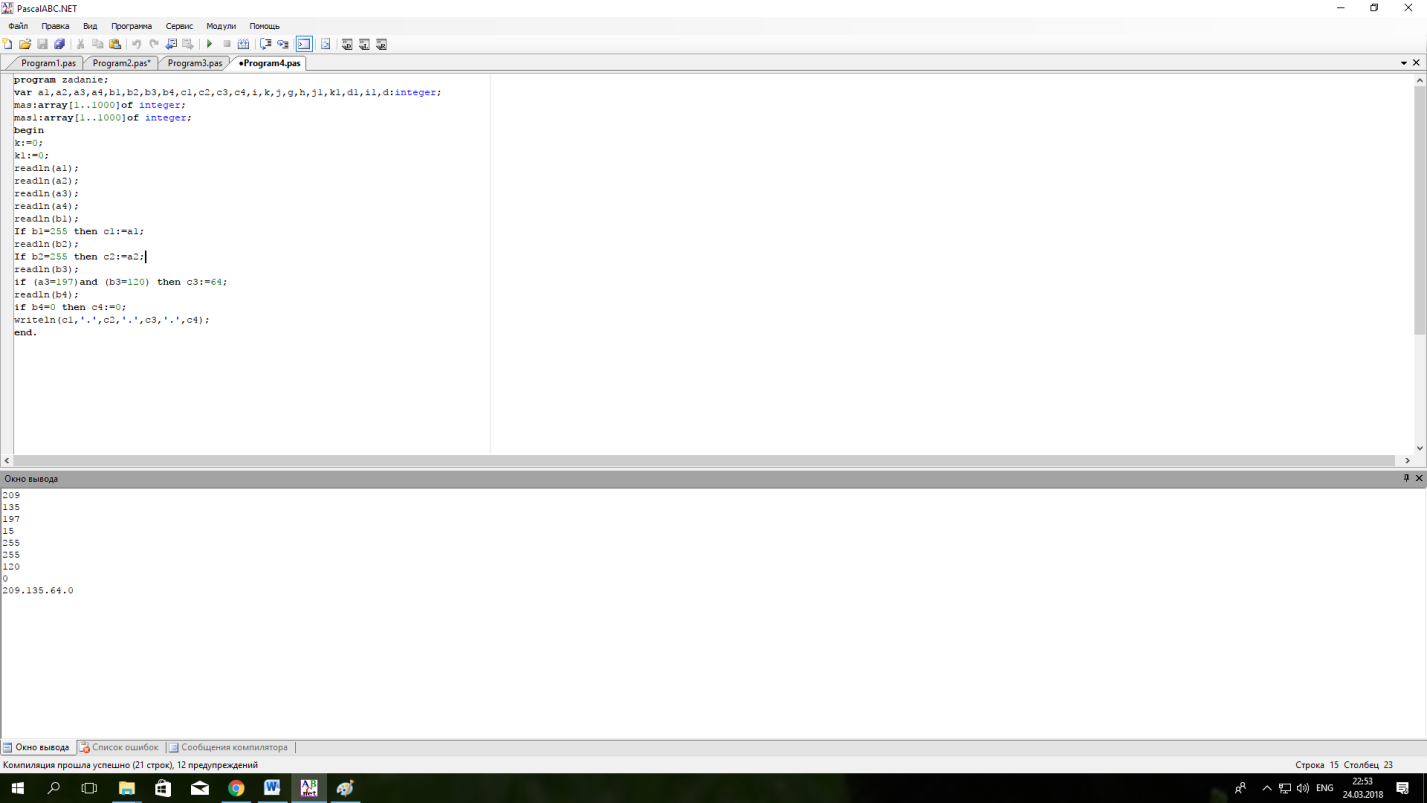
IP-адрес узла: 209.135.197.15

Маска подсети: 255.255.120.0

**Пример вывода:**

Сетевой адрес: 209.135.64.0





**Задание 5**

Разработать программу, которая определяет информационный объем всех результатов наблюдений влажности воздуха на метеорологической станции в байтах для некоторого фиксированного количества измерений. Результатом одного измерения влажности воздуха является целое число от 0 до 100 процентов, которое записывается при помощи минимально возможного количества бит.

**program** zadanie;

**var** a,b:real;

**begin**

readln(a);

b:=a\*7/8; т.к 2 в 7 степени превосходит число 100 поэтому переменную а умножаем на 7 и получаем количество бит,а так как ответ нужно выдать в байтах соответственно делим на 8

writeln(b);

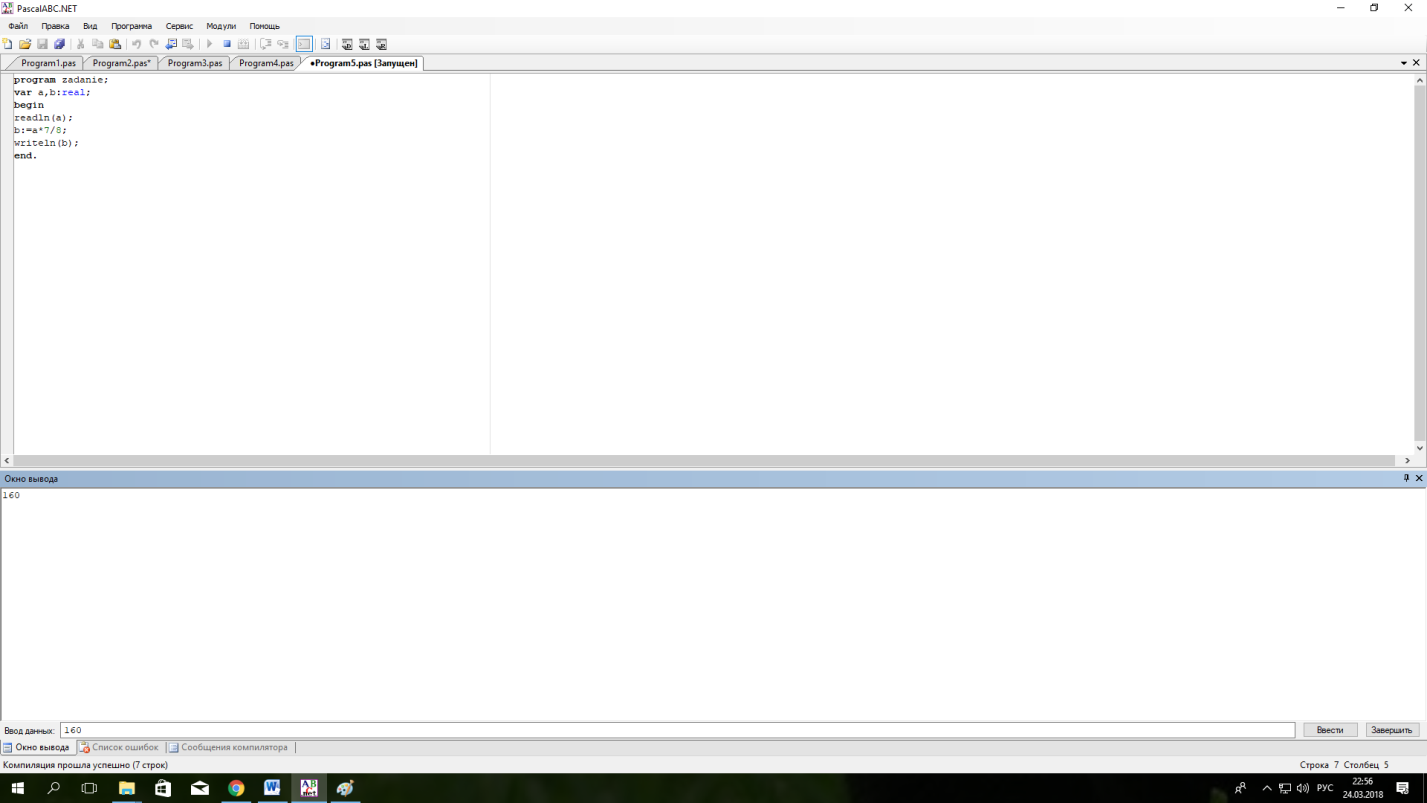
**end**.

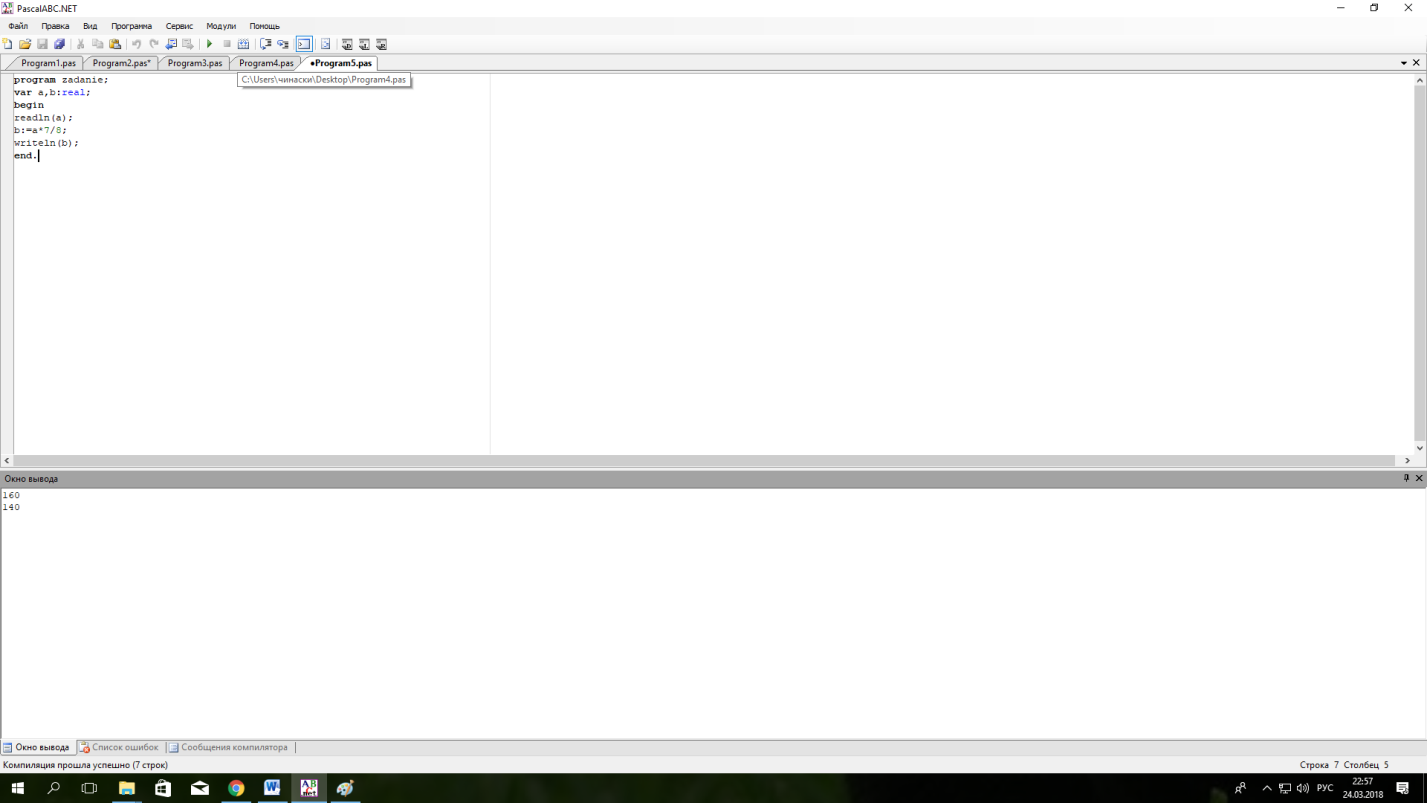
**Пример ввода:**

Количество измерений: 160

**Пример вывода:**

Информационный объем результатов наблюдений в байтах: 140





**Задание 6**

Разработать программу, которая подсчитывает количество нуклеотидов для заданной в программе строки. Подсчитать количество нуклеотидов A, C, G, T, разделив пробелами получаемые значения.

**program** zadanie;

**var** s:string;

i,k,k1,k2,k3:integer;

**begin**

k:=0;

k1:=0;

k2:=0;

k3:=0;

readln(s);

**for** i:=1 **to** length(s) **do**

**if** s[i]='A' **then** inc(k)**else**

**if** s[i]='C' **then** inc(k1)**else**

**if** s[i]='G' **then** inc(k2)**else**

**if** s[i]='T' **then** inc(k3);

write(k,',',k1,',',k2,',',k3);

**end**.

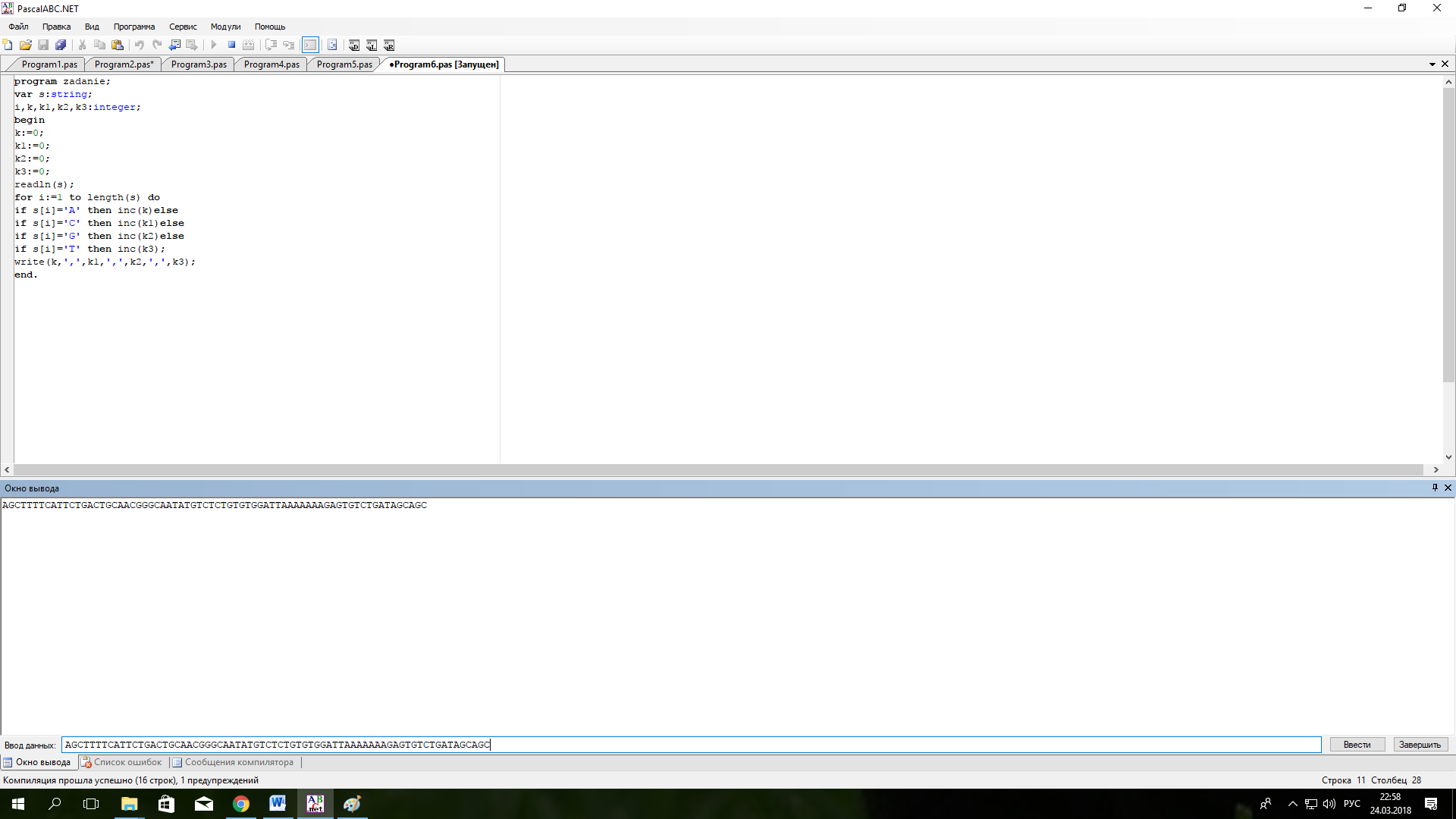
**Пример ввода:**

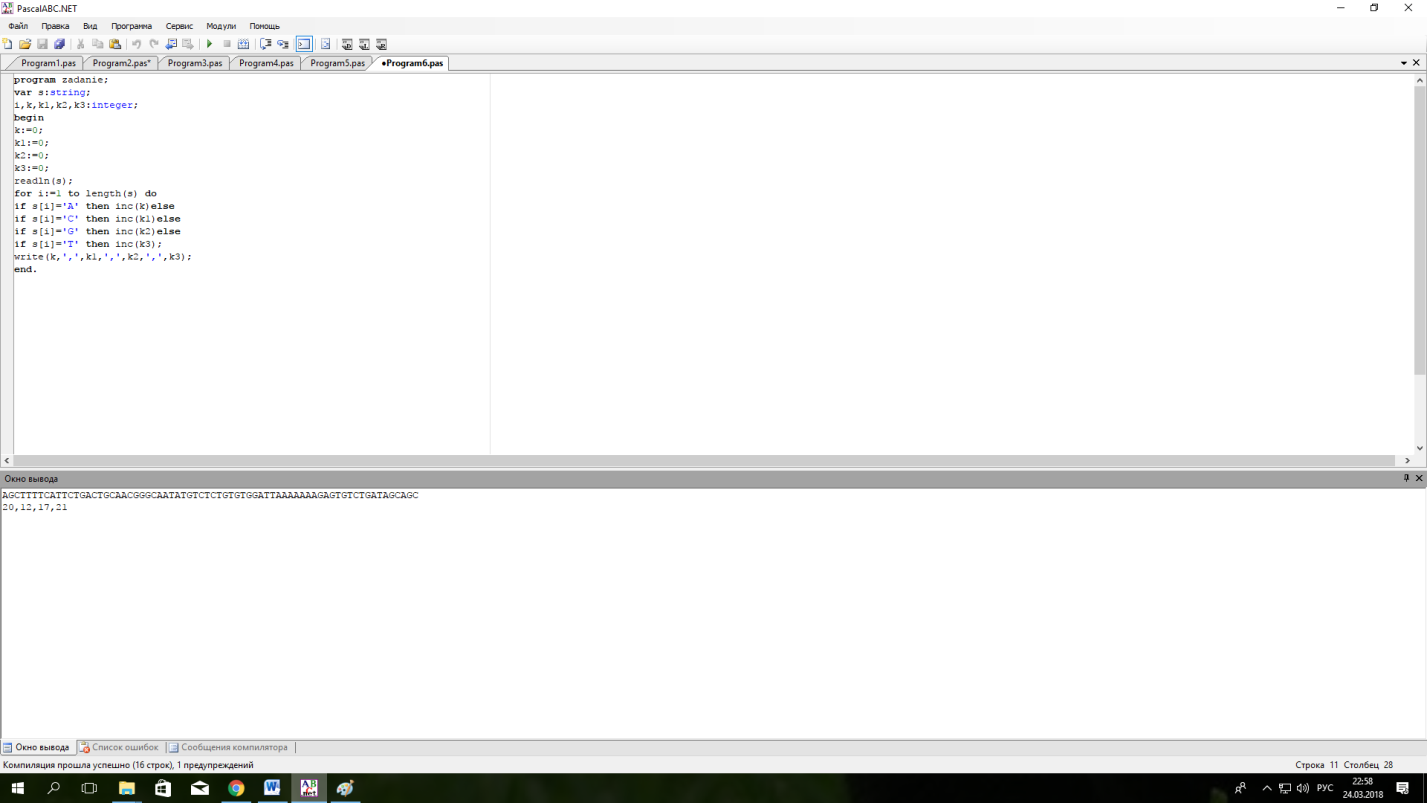
Дана половинка ДНК, в строковой переменной в программе присвоено значение последовательности нуклеотидов:

AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC

**Пример вывода:**

20 12 17 21





**Задание 7**

Разработать программу, которая выполняет построение комплементарной цепочки ДНК для заданной строки ДНК: AAAACCCGGT. Для построение комплементарной цепочки ДНК нужно прочитать цепочку задом наперед, заменив аденин на тимин, а цитозин на гуанин.

**program** zadanie;

**var**

s:string;

i,k,k1,k2,k3:integer;

**begin**

k:=0;

readln(s);

**for** i:=1 **to** length(s) **do begin**

**If** s[i]='A' **then** k:=k+1 **else**

**if** s[i]='C' **then** k1:=k1+1 **else**

**if** s[i]='G' **then** k2:=k2+1 **else**

**if** s[i]='T' **then** k3:=k3+1

**end**;

**for** k3:=1 **to** k3 **do**

write('A');

**for** k2:=1 **to** k2 **do**

write('C');

**for** k1:=1 **to** k1 **do**

write('G');

**for** k:=1 **to** k **do**

write('T');

**end**.

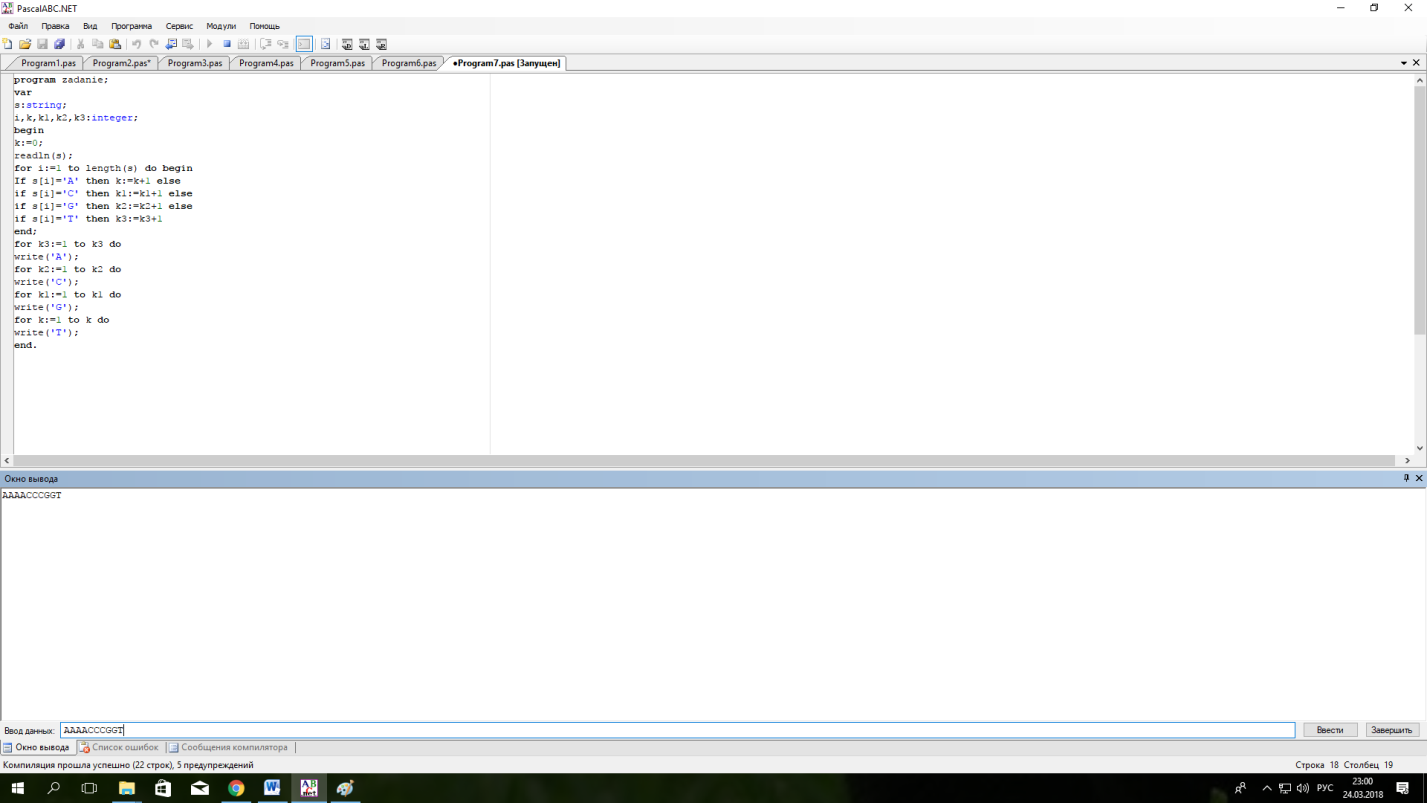
**Пример ввода:**

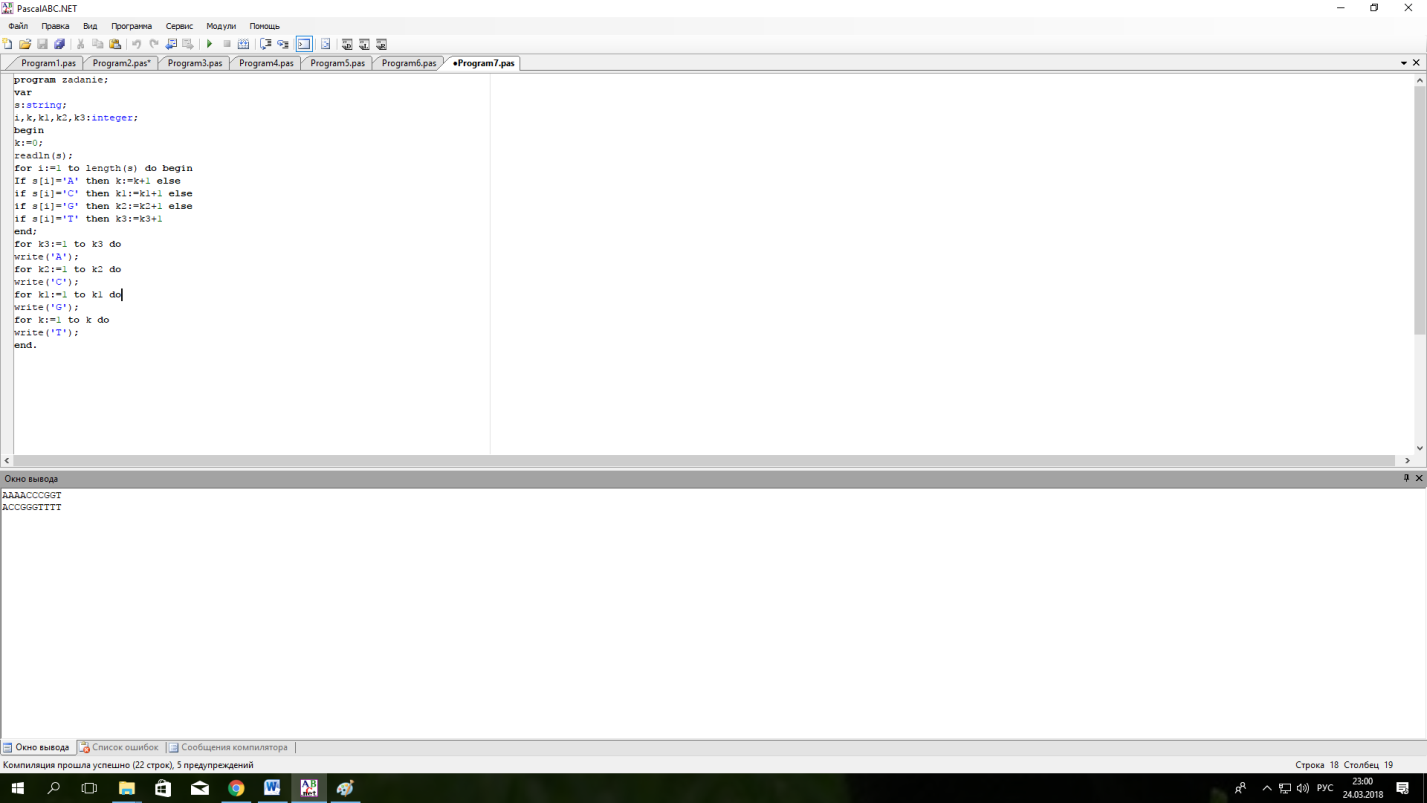
Строковой переменной в программе присвоено значение:

AAAACCCGGT

**Пример вывода:**

ACCGGGTTTT





**Задание 8**

Разработать программу, которая выполняет подсчет количества точечных мутаций.

Даны две строки:

GAGCCTACTAACGGGAT

CATCGTAATGACGGCCT

Требуется найти расстояние Хэмминга, то есть количество различий в символах – несовпадающих элементов в строках.

В примере нахождения расстояние Хэмминга несовпадающие элементы выделены красным цветом:

**program** zadanie;

**var**

s1,s2:string;

i,j,k,k1,k2,k3,l,l1,l2,l3,c:integer;

**begin**

readln(s1);

**for** i:=1 **to** length(s1) **do begin**

**If** s1[i]='A' **then** inc(k)**else**

**If** s1[i]='C' **then** inc(k1)**else**

**If** s1[i]='G' **then** inc(k2)**else**

**If** s1[i]='T' **then** inc(k3) ;

**end**;

readln(s2);

**for** j:=1 **to** length(s2) **do begin**

**If** s2[j]='A' **then** inc(l)**else**

**If** s2[j]='C' **then** inc(l1)**else**

**If** s2[j]='G' **then** inc(l2)**else**

**If** s2[j]='T' **then** inc(l3) ;

**end**;

**begin**

**If** k<>l **then** c:=c+abs(k-l)**else** c:=c+k;

**If** k1<>l1 **then** c:=c+abs(k1-l1)**else** c:=k1;

**If** k2<>l2 **then** c:=c+abs(k2-l2)**else** c:=k2;

**if** k3<>l3 **then** c:=c+abs(k3-l3)**else** c:=k3;

**end**;

writeln(c);

**end**.

**Пример ввода:**

Переменной Str1 в программе присвоено значение:

GAGCCTACTAACGGGAT

Переменной Str1 в программе присвоено значение:

CATCGTAATGACGGCCT

**Пример вывода:**

Расстояние Хэмминга: **7**

