Киреев Тимур Ильдарович

г. Уфа Лицей 107

timur\_kireev01@mail.ru

**Задание 1**

PascalABC.NET

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**.

**Var** f:byte; i,k,n:integer;

**Begin**

Write('F= ');

Read(f);

**if** (f >1) **or** (f<0) **then begin** writeln('Ошибка ввода');

**exit**;**end**;

**for** i:= 0 **to** 1 **do**

**for** k:=0 **to** 1 **do**

**for** n:=0 **to** 1 **do**

**if** ((k **or** n) **and not** i)= f **then begin**

Write(n,' ',k,' ' ,i); writeln; **end**; **end**.

**Пример ввода:**

1

**Пример вывода:**

 1 0 0

 0 1 0

 1 1 0



**Задание 2**

PascalABC.NET

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).

**Пример ввода:**

**20
20
20
20
90
90
90
90**

**Пример вывода:**

**Квадрат**



**Задание 4**

PascalABC.NET

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

**Var** a :**array**[1..4] **of** string;b :**array**[1..4] **of** string;c :**array** [1..4,1..8] **of** char;

d:**array** [1..4]**of** string;

i,i1,n:byte;vx,vx1,qw,wq:string;

**function** f(x:string):string;

**var** s:string;r,x1:integer; **begin**

val(x,x1,r); s:=''; **while** x1>0 **do begin**

s:=chr(ord('0')+x1 **mod** 2)+s; x1:=x1 **div** 2; **end**;

**for** x1:=1 **to** 8-length(s) **do** s:='0'+s; f:=s; **end**;

**function** j(x:string):string;

**var** s:string;i:integer;d:real; **begin** d := 0;

**for** i := 0 **to** length(x) - 1 **do begin**

**if** x[i+1] = '1' **then** d := d +exp(ln(2)\*((length(x)-1) - i));

**end**;str(d,s); j:=s;**end**; **Begin**

Write('IP-адрес узла: '); readln(vx); Write('Маска подсети: '); readln(vx1);

**for** i:=1 **to** 4 **do begin**

n:=pos('.',vx); a[i]:=copy (vx,1,n-1); delete(vx,1,n);**end**;

a[4]:=copy (vx,1,length(vx)); **for** i:=1 **to** 4 **do begin**

n:=pos('.',vx1); b[i]:=copy (vx1,1,n-1); delete(vx1,1,n);**end**;

b[4]:=copy (vx1,1,length(vx1)); **for** i:=1 **to** 4 **do begin**

a[i]:=f(a[i]); b[i]:=f(b[i]); qw:=a[i];wq:=b[i]; **for** i1:=1 **to** 8 **do begin**

**if** (qw[i1]='1') **and** (wq[i1]='1') **then** c[i,i1]:='1' **else** c[i,i1]:='0'; **end**;**end**;

**for** i:=1 **to** 4 **do for** i1:=1 **to** 8 **do**

d[i]:=d[i]+c[i,i1]; write('Сетевой адрес: ');

**for** i:=1 **to** 4 **do begin** d[i]:=j(d[i]);

**if** i<>4 **then** write(d[i],'.') **else** write (d[i]); **end**; **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**

PascalABC.NET

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.

**Var** k,x:integer;a:real;

**Begin**

Write('Количество измерений: ');

Readln(x);

a:=x\*7/8;

k:=trunc(a);

**if** x\*7 **mod** 8 <> 0 **then** k:=k+1;

Writeln('Информационный объем результатов наблюдений в байтах: ',k);

**end**.

**Пример ввода:**

160

**Пример вывода:**
140



**Задание 6**

PascalABC.NET

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

**Var** str:string;k,i,a,g,c,t:integer;

**Begin**

Writeln('Введите строку');

Readln(str);

**For** i:=1 **to** length(str) **do begin**

**if** str[i]='A' **then** a:=a+1;

**if** str[i]='G' **then** g:=g+1;

**if** str[i]='T' **then** t:=t+1;

**if** str[i]='C' **then** c:=c+1;

**end**;

writeln('A= ',a);

writeln('C= ',c);

writeln('G= ',g);

writeln('T= ',t);

**end**.

**Пример ввода:**

AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGTGTCTGATAGCAGC

**Пример вывода:**

A= 20

C= 12

G= 17

T= 21



**Задание 7**

PascalABC.NET

**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.**

**Var** str,str1:string;i:integer;

**Begin**

Writeln('Введите строку');

Readln(str);

**For** i:=1 **to** length(str) **do begin**

str1:=str[i]+str1;

**end**;

**For** i:=1 **to** length(str1) **do begin**

**if** str1[i]='A' **then** str1[i]:='T' **else**

**if** str1[i]='T' **then** str1[i]:='A' **else**

**if** str1[i]='G' **then** str1[i]:='C' **else**

**if** str1[i]='C' **then** str1[i]:='G';

**end**;

writeln(str1);

**end**.

**Пример ввода:**

**AAAACCCGGT**

**Пример вывода:**

**ACCGGGTTTT**



**Задание 8**

PascalABC.NET

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.

**Var** str1,str2:string;i:integer;k:byte;

**Begin**

Writeln('Введите первую строку ');

Readln(str1);

Writeln('Введите вторую строку ');

Readln(str2);

**For** i:=1 **to** length(str1) **do begin**

**if** str1[i]<>str2[i] **then** k:=k+1;

**end**;

Writeln('Расстояние Хэмминга: ',k);

**end**.

**Пример ввода:**

GAGCCTACTAACGGGAT

CATCGTAATGACGGCCT

**Пример вывода:**

Расстояние Хэмминга: **7**

