|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ | | | | | | | | | | | | |
| Федеральное государственное автономное образовательное учреждение  высшего образования | | | | | | | | | | | | |
| **Дальневосточный федеральный университет** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **Департамент программной инженерии и искусственного интеллекта** | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **О Т Ч Е Т** | | | | | | | | | | | | |
| по лабораторной работе №4  дисциплина «Основы алгоритмизации» | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | |  |  | Студент гр. Б9121-02.03.03тп | | | |
|  |  |  | | А.В. Плахотный | |
|  | | | | | | |  |  | (подпись) | |  | |
|  | | | | | | | | | | | | |
|  | | | | | | |  |  | Руководитель | | |  |
|  | | | | | | |  |  | ст. преподаватель | | | |
|  | | | |  |  | |  |  |  |  | С.Н. Остроухова | |
|  | | | |  |  | |  |  | (подпись) |  | (И.О. Фамилия) | |
|  |  |  |  | | |  |  |  |  | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| г. Владивосток | | | | | | | | | | | | |
| 2021 | | | | | | | | | | | | |

**Программа содержащая максимальное количество**

**логических выражений.**

В текстовом файле содержится 3 программы на языке паскаль (Программа от программы отделяется любым количеством пробелов и строк (Возможно 3 программы записаны в одну строчку)). Вывести во второй файл программу, которая содержит максимальное количество логических выражений.

Во входных программах не будет строк (никаких), не будет комментариев.

Начало программ определять не при помощи end. , а при помощи program . Логическое выражение. Выражение выдающее логическое значение. Например: if … then, while…do, repeat…until.

**2 Неформальная постановка задачи (НПЗ)**

В текстовом файле содержится 3 программы на языке паскаль (Программа от программы отделяется любым количеством пробелов и строк (Возможно 3 программы записаны в одну строчку)). Вывести во второй файл программу, которая содержит максимальное количество логических выражений.

Во входных программах не будет строк (никаких), не будет комментариев.

Логическое выражение. Выражение - выдающее логическое значение. Например: if … then, while…do, repeat…until.

**2 Формальная постановка задачи (ФПЗ)**

X:

In1 – текстовый файл который содержит 3 программы на языке pascal разделенные между собой произвольным числом строк и пробелов

Out – пустой текстовый файл

Y:

Out – текстовый файл содержащий одну из трех программ на языке pascal файла in1, а в частности программу на языке pascal содержащую наибольшее количество логических выражений.

Вывод в компиляторе трех строк в следующей форме:

Количество логических выражений в программе 1 равно x

Количество логических выражений в программе 2 равно y

Количество логических выражений в программе 3 равно z

Где {x, y, z} - принадлежат ряду натуральных чисел (N)

F:

Считаем количество ‘if ‘, ‘ while ‘ и ‘repeat’ в программах на языке pascal файла in1. Начало программ определять не при помощи end. , а при помощи program .

Копируем программы на языке pascal из файла in1 в массив. Вставляем в текстовый файл out программу на языке pascal содержащую наибольшее количество ‘if ‘, ‘ while ‘ и ‘repeat’.

**3 Текст программы**

**const**

num = 5;

**var**

in1,outp:text;

s,slovo:string;

c: boolean;

k:char;

max,stroka,prog,pr,i,j,w:integer;

mas: **array** [1..num,1..num] **of** integer;

**begin**

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

**for** j:= 1 **to** num **do**

mas[i,j]:=0;

assign(in1,'in6.txt');

assign(outp,'out.txt');

reset(in1);

rewrite(outp);

**while not** eof(in1) **do**

**begin**

readln(in1,s);

slovo:= '';

inc(stroka);

s:= s + ' ';

c:=false;

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

**begin**

**case** s[i] **of**

'=': **begin if** s[i-1]=':'

**then** c:=true

**else if** c= true

**then begin** inc(mas[prog,3]);

c:=false;

**end**;

**end**;

'>','<':**begin if** c=true

**then begin** inc(mas[prog,3]);

c:=false;

**end**;

**end**;

' ':**begin**

**if** s <> '' **then**

**begin**

**if** slovo = 'program' **then**

**begin**

inc(prog);

mas[prog, 1] := stroka;

mas[prog, 2] := pos('program', s);

slovo:='';

**end**

**else begin**

slovo:=slovo+' ';

**if** slovo = 'if ' **then**

inc(mas[prog,3]);

**if** slovo = 'while ' **then**

inc(mas[prog,3]);

**if** slovo = 'repeat ' **then**

inc(mas[prog,3]);

slovo:= '';

**end**;

**end**;

**end**;

';': c:=false

**else begin** slovo:= slovo + s[i] **end**;

**end**;

**end**;

**end**;

**for** i := 1 **to** 3 **do**

writeln('Количество логических выражений в программе ', i, ' равно ', mas[i, 3]);

max := mas[1, 3];

pr := 1;

**if** mas[2, 3] > max **then begin**

max := mas[2, 3];

pr := 2;

**end**;

**if** mas[3, 3] > max **then begin**

max := mas[3, 3];

pr := 3;

**end**;

close(in1);

close(outp);

reset(in1);

rewrite(outp);

stroka:=0;

**while not** eof(in1) **do**

**begin**

readln(in1,s);

inc(stroka);

**if** pr =3 **then begin**

**if** stroka = mas[3,1] **then**

writeln(outp,copy(s,mas[3,2],length(s)-mas[3,2]))

**else if** stroka>mas[3,1] **then**

writeln(outp,s);

**end**;

**if** pr =2 **then begin**

**if** (stroka = mas[2,1]) **and** (mas[2,1] = mas[1,1])

**and** (mas[2,1]=mas[3,1]) **then**

writeln(outp,copy(s,mas[2,2],mas[2,2]-mas[3,2]))

**else if** (stroka = mas[2,1]) **and** (mas[2,1] = mas[1,1]) **then**

writeln(outp,copy(s,mas[2,2],length(s)-mas[2,2]))

**else if** (stroka = mas[2,1]) **and** (mas[2,1] = mas[3,1]) **then**

writeln(outp,copy(s,mas[2,2],mas[2,2]-mas[3,2]))

**else if** stroka= mas[2,1] **then**

writeln(outp,copy(s,mas[2,2],length(s)))

**else if** (stroka> mas[2,1])**and** (stroka<mas[3,1]) **then**

writeln(outp,s)

**else if** stroka = mas[3,1] **then**

writeln(outp,copy(s,1,mas[3,2]-1))

**end**;

**if** pr=1 **then begin**

**if** (stroka = mas[1,1]) **and** (mas[1,1] = mas[2,1]) **then**

writeln(outp,copy(s,mas[1,1],mas[2,2]-mas[1,2]))

**else if** (stroka>=mas[1,1])**and**(stroka<mas[2,1]) **then**

writeln(outp,s)

**else if** stroka = mas[2,1] **then**

writeln(outp,copy(s,1,mas[2,2]-1))

**end**;

**end**;

close(in1);

close(outp);

**end**.

[**https://onlinegdb.com/XW-wl6tGn**](https://onlinegdb.com/XW-wl6tGn)

**4 Тесты**

|  |  |  |
| --- | --- | --- |
| № | Тестовая ситуация | Вывод |
|
| 1 | 3 тестовые программы записаны в одну строку (in2) | Количество логических выражений в программе 1 равно 12  Количество логических выражений в программе 2 равно 16  Количество логических выражений в программе 3 равно 4 |
| 2 | У каждой тестовой программы имеется своя строка(in3) | Количество логических выражений в программе 1 равно 10  Количество логических выражений в программе 2 равно 16  Количество логических выражений в программе 3 равно 4 |
| 3 | 3 тестовые программы записаны в нормальном виде(in1) | Количество логических выражений в программе 1 равно 10  Количество логических выражений в программе 2 равно 16  Количество логических выражений в программе 3 равно 4 |
| 4 | 3 программы записаны в нормальном виде(in4) | Количество логических выражений в программе 1 равно 16  Количество логических выражений в программе 2 равно 13  Количество логических выражений в программе 3 равно 13 |
| 5 | У каждой программы имеется своя строка(in6) | Количество логических выражений в программе 1 равно 16  Количество логических выражений в программе 2 равно 13  Количество логических выражений в программе 3 равно 13 |
| 6 | 3 программы записаны в одну строку (in5) | Количество логических выражений в программе 1 равно 16  Количество логических выражений в программе 2 равно 13  Количество логических выражений в программе 3 равно 13 |

**4 Тесты**

|  |  |  |
| --- | --- | --- |
| № | Тестовая ситуация | Вывод |
|
| 1 | program  if a  if a  if a repeat a repeat a  a:=j=k;  h := (f>b) and (6  <  7) or (6<7)false;  repeat a  if a  if a  if a  ifa  whilea  repeat a  program fuuu  if a  if a if a if a ifa if if a  if a  if a while a while a  while a  if a  if a  if a  if a  a:=w>r  aaaaaa a  program  aaaa if a if a if a gif a  aaa a if a  ifed aaa  aaa | program fuuu  if a  if a if a if a ifa if if a  if a  if a while a while a  while a  if a  if a  if a  if a  a:=w>r  aaaaaa a |
| 2 | program if a if a if a repeat a repeat a repeat a if a if a if a ifa whilea repeat a program fuuu if a if a if a if a if a if if a if a if a while a while a while a if a if a if a if a aaaaaa a program aaaa if a if a if a gif a aaa a if a ifed aaa aaa | program fuuu if a if a if a if a if a if if a if a if a while a while a while a if a if a if a if a aaaaaa a |
| 3 | program if a if a if a repeat a repeat a repeat a if a if a if a ifa whilea repeat a  program fuuu if a if a if a if a if a if if a if a if a while a while a while a if a if a if a if a aaaaaa a  program aaaa if a if a if a gif a aaa a if a ifed aaa aaa | program fuuu if a if a if a if a if a if if a if a if a while a while a while a if a if a if a if a aaaaaa a |
| 4 | program Example\_1;  uses GraphABC;  var x0,y0, alfa:integer;  len, d\_len:integer;  i:integer;  c:color;  f\_close, f\_fill:boolean;  e55E:integer;  num\_program:byte;  num\_prog:byte;  begin  e55E:=i+e55e+4;  SetWindowSize(1000,600);  SetPenWidth(5);  x0:=WindowWidth div 2;  y0:=WindowHeight div 2;  SetCoordinateOrigin(x0,y0);  alfa:=0;  len:=100;  d\_len:=20;  i:=0;  f\_close:=true;  f\_fill:=false;  if 0<num\_prog  num\_program:=num\_program+num\_prog  while f\_close do begin  i:=i+1;  if i mod 2=0  then SetPenColor(clGreen)  else SetPenColor(clOrange);  DrawRectangle(0,0,len,len);  sleep(500);  ClearWindow;  alfa:=alfa+30;  SetCoordinateAngle(alfa);  OnKeyDown:=StopProgram;  end;  end.  program Example\_2;  uses GraphABC;  var x0,y0, alfa:integer;  len, d\_len:integer;  i:integer;  c:color;  f\_close, f\_fill:boolean;  begin  SetWindowSize(1000,600);  SetPenWidth(5);  x0:=WindowWidth div 2;  y0:=WindowHeight div 2;  SetCoordinateOrigin(x0,y0);  alfa:=0;  len:=100;  d\_len:=20;  i:=0;  f\_close:=true;  f\_fill:=false;  while f\_close do begin  i:=i+1;  if i mod 2=0  then SetPenColor(clGreen)  else SetPenColor(clOrange);  DrawRectangle(0,0,len,len);  sleep(500);  ClearWindow;  alfa:=alfa+30;  SetCoordinateAngle(alfa);  OnKeyDown:=StopProgram;  end;  end.  program Example\_3;  uses GraphABC;  var x0,y0, alfa:integer;  len, d\_len:integer;  i:integer;  c:color;  f\_close, f\_fill:boolean;  begin  SetWindowSize(1000,600);  SetPenWidth(5);  x0:=WindowWidth div 2;  y0:=WindowHeight div 2;  SetCoordinateOrigin(x0,y0);  alfa:=0;  len:=100;  d\_len:=20;  i:=0;  f\_close:=true;  f\_fill:=false;  while f\_close do begin  i:=i+1;  if i mod 2=0  then SetPenColor(clGreen)  else SetPenColor(clOrange);  DrawRectangle(0,0,len,len);  sleep(500);  ClearWindow;  alfa:=alfa+30;  SetCoordinateAngle(alfa);  OnKeyDown:=StopProgram;  end;  end. | program Example\_1;  uses GraphABC;  var x0,y0, alfa:integer;  len, d\_len:integer;  i:integer;  c:color;  f\_close, f\_fill:boolean;  e55E:integer;  num\_program:byte;  num\_prog:byte;  begin  e55E:=i+e55e+4;  SetWindowSize(1000,600);  SetPenWidth(5);  x0:=WindowWidth div 2;  y0:=WindowHeight div 2;  SetCoordinateOrigin(x0,y0);  alfa:=0;  len:=100;  d\_len:=20;  i:=0;  f\_close:=true;  f\_fill:=false;  if 0<num\_prog  num\_program:=num\_program+num\_prog  while f\_close do begin  i:=i+1;  if i mod 2=0  then SetPenColor(clGreen)  else SetPenColor(clOrange);  DrawRectangle(0,0,len,len);  sleep(500);  ClearWindow;  alfa:=alfa+30;  SetCoordinateAngle(alfa);  OnKeyDown:=StopProgram;  end;  end. |
| 5 | program Example\_1; uses GraphABC; var x0,y0, alfa:integer; len, d\_len:integer; i:integer; c:color; f\_close, f\_fill:boolean; e55E:integer; num\_program:byte; num\_prog:byte; begine55E:=i+e55e+4; SetWindowSize(1000,600); SetPenWidth(5); x0:=WindowWidth div 2; y0:=WindowHeight div 2; SetCoordinateOrigin(x0,y0); alfa:=0; len:=100; d\_len:=20; i:=0; f\_close:=true; f\_fill:=false; if 0<num\_prognum\_program:=num\_program+num\_prog while f\_close do begini:=i+1; if i mod 2=0then SetPenColor(clGreen)else SetPenColor(clOrange); DrawRectangle(0,0,len,len); sleep(500); ClearWindow; alfa:=alfa+30; SetCoordinateAngle(alfa); OnKeyDown:=StopProgram; end; end. program Example\_2; uses GraphABC; var x0,y0, alfa:integer; len, d\_len:integer; i:integer; c:color; f\_close, f\_fill:boolean; beginSetWindowSize(1000,600); SetPenWidth(5); x0:=WindowWidth div 2; y0:=WindowHeight div 2; SetCoordinateOrigin(x0,y0); alfa:=0; len:=100; d\_len:=20; i:=0; f\_close:=true; f\_fill:=false; while f\_close do begini:=i+1; if i mod 2=0then SetPenColor(clGreen)else SetPenColor(clOrange); DrawRectangle(0,0,len,len); sleep(500); ClearWindow; alfa:=alfa+30; SetCoordinateAngle(alfa); OnKeyDown:=StopProgram; end; end. program Example\_3; uses GraphABC; var x0,y0, alfa:integer; len, d\_len:integer; i:integer; c:color; f\_close, f\_fill:boolean; beginSetWindowSize(1000,600); SetPenWidth(5); x0:=WindowWidth div 2; y0:=WindowHeight div 2; SetCoordinateOrigin(x0,y0); alfa:=0; len:=100; d\_len:=20; i:=0; f\_close:=true; f\_fill:=false; while f\_close do begini:=i+1; if i mod 2=0then SetPenColor(clGreen)else SetPenColor(clOrange); DrawRectangle(0,0,len,len); sleep(500); ClearWindow; alfa:=alfa+30; SetCoordinateAngle(alfa); OnKeyDown:=StopProgram; end; end. | program Example\_1; uses GraphABC; var x0,y0, alfa:integer; len, d\_len:integer; i:integer; c:color; f\_close, f\_fill:boolean; e55E:integer; num\_program:byte; num\_prog:byte; begine55E:=i+e55e+4; SetWindowSize(1000,600); SetPenWidth(5); x0:=WindowWidth div 2; y0:=WindowHeight div 2; SetCoordinateOrigin(x0,y0); alfa:=0; len:=100; d\_len:=20; i:=0; f\_close:=true; f\_fill:=false; if 0<num\_prognum\_program:=num\_program+num\_prog while f\_close do begini:=i+1; if i mod 2=0then SetPenColor(clGreen)else SetPenColor(clOrange); DrawRectangle(0,0,len,len); sleep(500); ClearWindow; alfa:=alfa+30; SetCoordinateAngle(alfa); OnKeyDown:=StopProgram; end; end. |
| 6 | program Example\_1; uses GraphABC; var x0,y0, alfa:integer; len, d\_len:integer; i:integer; c:color; f\_close, f\_fill:boolean; e55E:integer; num\_program:byte; num\_prog:byte; begine55E:=i+e55e+4; SetWindowSize(1000,600); SetPenWidth(5); x0:=WindowWidth div 2; y0:=WindowHeight div 2; SetCoordinateOrigin(x0,y0); alfa:=0; len:=100; d\_len:=20; i:=0; f\_close:=true; f\_fill:=false; if 0<num\_prognum\_program:=num\_program+num\_prog while f\_close do begini:=i+1; if i mod 2=0then SetPenColor(clGreen)else SetPenColor(clOrange); DrawRectangle(0,0,len,len); sleep(500); ClearWindow; alfa:=alfa+30; SetCoordinateAngle(alfa); OnKeyDown:=StopProgram; end; end.  program Example\_2; uses GraphABC; var x0,y0, alfa:integer; len, d\_len:integer; i:integer; c:color; f\_close, f\_fill:boolean; beginSetWindowSize(1000,600); SetPenWidth(5); x0:=WindowWidth div 2; y0:=WindowHeight div 2; SetCoordinateOrigin(x0,y0); alfa:=0; len:=100; d\_len:=20; i:=0; f\_close:=true; f\_fill:=false; while f\_close do begini:=i+1; if i mod 2=0then SetPenColor(clGreen)else SetPenColor(clOrange); DrawRectangle(0,0,len,len); sleep(500); ClearWindow; alfa:=alfa+30; SetCoordinateAngle(alfa); OnKeyDown:=StopProgram; end; end.  program Example\_3; uses GraphABC; var x0,y0, alfa:integer; len, d\_len:integer; i:integer; c:color; f\_close, f\_fill:boolean; beginSetWindowSize(1000,600); SetPenWidth(5); x0:=WindowWidth div 2; y0:=WindowHeight div 2; SetCoordinateOrigin(x0,y0); alfa:=0; len:=100; d\_len:=20; i:=0; f\_close:=true; f\_fill:=false; while f\_close do begini:=i+1; if i mod 2=0then SetPenColor(clGreen)else SetPenColor(clOrange); DrawRectangle(0,0,len,len); sleep(500); ClearWindow; alfa:=alfa+30; SetCoordinateAngle(alfa); OnKeyDown:=StopProgram; end; end. | program Example\_1; uses GraphABC; var x0,y0, alfa:integer; len, d\_len:integer; i:integer; c:color; f\_close, f\_fill:boolean; e55E:integer; num\_program:byte; num\_prog:byte; begine55E:=i+e55e+4; SetWindowSize(1000,600); SetPenWidth(5); x0:=WindowWidth div 2; y0:=WindowHeight div 2; SetCoordinateOrigin(x0,y0); alfa:=0; len:=100; d\_len:=20; i:=0; f\_close:=true; f\_fill:=false; if 0<num\_prognum\_program:=num\_program+num\_prog while f\_close do begini:=i+1; if i mod 2=0then SetPenColor(clGreen)else SetPenColor(clOrange); DrawRectangle(0,0,len,len); sleep(500); ClearWindow; alfa:=alfa+30; SetCoordinateAngle(alfa); OnKeyDown:=StopProgram; end; end. |

var

start\_hour, stop\_hour, start\_minute, stop\_minute: integer;

interval: string; //задается пользователем, интервал подсчёта телепрограмм

function format(vremya: string):string; //функция для перевода из 12 часового формата в 24 часовой

var

temp: string;

begin

if 'a.m.' in vremya then begin //работа с a.m.

temp := vremya.ToWords('a')[0];

if temp.ToWords(':')[0].ToInteger <> 12 then begin

format := '0' + vremya.ToWords('a')[0];

end

else begin

format := '00:' + (vremya.ToWords('a')[0].ToWords(':')[1]).ToString;

end;

end;

if 'p.m.' in vremya then begin //работа с p.m.

temp := vremya.ToWords('p')[0];

if temp.ToWords(':')[0].ToInteger <> 12 then begin

format := (vremya.ToWords('p')[0].ToWords(':')[0].ToInteger + 12).ToString + ':' + vremya.ToWords('p')[0].ToWords(':')[1];

end

else begin

format := (vremya.ToWords('p')[0]);

end;

end;

if not ('a.m.' in vremya) and not ('p.m.' in vremya) then begin //если не a.m. и не p.m.

format := vremya;

end;

end;

procedure Channels(); //обработка файла

var

f,output: text;

channel, line: string;

count: integer;

was\_count: boolean;

begin

assign(f, 'channels.txt'); //база с каналами

reset(f);

assign(output, 'output.txt'); //очистка файла с ответами

rewrite(output);

close(output);

assign(output, 'output.txt'); //на дозапись

append(output);

start\_hour := interval.ToWords('-')[0].ToWords(':')[0].ToInteger; //разбитие введенных часов для поиска от пользователя

stop\_hour := interval.ToWords('-')[1].ToWords(':')[0].ToInteger;

start\_minute := interval.ToWords('-')[0].ToWords(':')[1].ToInteger;

stop\_minute := interval.ToWords('-')[1].ToWords(':')[1].ToInteger;

while not eof(f) do

begin

readln(f, line);

channel := line.ToWords(' ')[0];

count := 0;

for var i := 1 to length(line.ToWords(' ')) - 1 do //проверка телепередач в цикле

begin

was\_count := False;

if not (was\_count) and ((format(line.ToWords(' ')[i])).ToWords(':')[0].ToInteger >= start\_hour) and ((format(line.ToWords(' ')[i])).ToWords(':')[0].ToInteger < stop\_hour) and ((format(line.ToWords(' ')[i])).ToWords(':')[1].ToInteger >= start\_minute) then begin

was\_count := True;

inc(count);

end;

if not (was\_count) and ((format(line.ToWords(' ')[i])).ToWords(':')[0].ToInteger > start\_hour) and ((format(line.ToWords(' ')[i])).ToWords(':')[0].ToInteger <= stop\_hour) and ((format(line.ToWords(' ')[i])).ToWords(':')[1].ToInteger <= stop\_minute) then begin

was\_count := True;

inc(count);

end;

if not (was\_count) and ((format(line.ToWords(' ')[i])).ToWords(':')[0].ToInteger = start\_hour) and ((format(line.ToWords(' ')[i])).ToWords(':')[0].ToInteger = stop\_hour) and ((format(line.ToWords(' ')[i])).ToWords(':')[1].ToInteger >= start\_minute) and ((format(line.ToWords(' ')[i])).ToWords(':')[1].ToInteger <= stop\_minute) then begin

was\_count := True;

inc(count);

end;

if not (was\_count) and (start\_hour > stop\_hour) then begin

if not (was\_count) and ((format(line.ToWords(' ')[i])).ToWords(':')[0].ToInteger = start\_hour) and ((format(line.ToWords(' ')[i])).ToWords(':')[1].ToInteger >= start\_minute) then begin

was\_count := True;

inc(count);

end;

if not (was\_count) and ((format(line.ToWords(' ')[i])).ToWords(':')[0].ToInteger = stop\_hour) and ((format(line.ToWords(' ')[i])).ToWords(':')[1].ToInteger <= stop\_minute) then begin

was\_count := True;

inc(count);

end;

if not (was\_count) and ((format(line.ToWords(' ')[i])).ToWords(':')[0].ToInteger > start\_hour) then begin

was\_count := True;

inc(count);

end;

if not (was\_count) and ((format(line.ToWords(' ')[i])).ToWords(':')[0].ToInteger < stop\_hour) then begin

was\_count := True;

inc(count);

end;

end;

end;

writeln(output,'В канале ',channel,' ', count, ' передач подходят под заданный поиск.');

end;

close(f);

close(output);

writeln;

writeln('Ответ записан в файл output.txt');

end;

begin

writeln('Задайте интервал времени в 24-х часовом формате. Например: 16:40-22:35');

readln(interval);

Channels();

writeln('Программа завершена.')

end.