Министерство путей сообщения Российской Федерации

Дальневосточный государственный университет

путей сообщения

Кафедра ”Информационные технологии и системы **”**

# Курсовая работа по информатике

## **Вариант № 9**

Выполнил: ст. 419г. Киршев И. Ф.

Проверил:

Березнев Д. П.

1998

**Составить программу определения минимального числа цветов, необходимых для раскраски карты произвольной конфигурации таким образом, чтобы страны с одинаковой раскраской не соприкасались. Схему границ карты представить массивом. На внешних файлах расположить 3 - 4 схемы расположения стран. Итоги представить в виде текста с указанием выбранных для каждой из стран цветов. Желательно завершить программу графическим приложением.**

**Переменные:**

"num" - номер файла, выбираемый пользователем.

"filen" - имя файла.

"g[1..100] - массив, используемый "генератором перебора всех вариантов"

"s:array[i,j]" - массив "связей" показывает, есть ли связь между странами "i" и "j".

"n" - количество цветов, используемых для раскраски.

"max - максимально возможное количество стран (определяется при считывании данных).

"s1,s2,k,j,i,a" - переменные, для работы "генератора".

"f, f1" - переменные для работы с файлами.

**"function get:integer;"**

Функция считывает текущее число, из файла связанного с переменной - "f" и ищет максимальное число из всех считанных.

Пока строка = '' или символ является:

- цифрой,

- "-",

- "."

считывает символ.

Если символ является:

- цифрой,

- "-",

- ".",

то он добавляется в строку "s".

Строка цифр "s" преобразуется в число. Если "max" меньше числа, то "max" приравнивается считанному числу.

**"function pr:boolean;"**

Функция проверяет, можно ли страну - i закрасить цветом - g[i] (Можно ли углубляться по дереву).

Перебирает все раскрашенные страны (от "1" до "i-1") и сравнивает цвета каждой из них с цветом страны "i".

**"function gen(n:integer):boolean;"**

Функция, определяющая возможность раскраски стран "n"-ым кол-вом цветов.

Каждому элементу массива "g" присваивает значение равное "0". Текущему номеру рассматриваемой страны "i" задает значение "1".

Повторяет действия:

Повторяет действие:

### К номеру цвета рассматриваемой страны прибавить "1" ("g[i]:=g[i]+1;"), пока нельзя страну "i" раскрасить в цвет "g[i]" или цвет "i"-ой страны не больше числа "n". Если цвет "i"-ой страны больше числа "n" то:

### номеру цвета рассматриваемой

страны приравнивает "0" и

понижает номер рассматриваемой страны на "1".

Иначе повышает номер рассматриваемой страны на "1", пока номер рассматриваемой страны не равен "1" или номер рассматриваемой страны не больше количества стран.

### Значение функции равно "True", если номер рассматриваемой страны больше количества стран.

**Основная программа:**

## Вывод сообщений пользователю (см. рабочую программу)

## Запрос номера файла ("num").

Выполняются действия в переменную "num" запрашивается символ нажатой клавиши, если нажата клавиша не от 1 до 3 то выводится сообщение об ошибке пока не нажата клавиша от 1 до 3.

Формируется имя исходного файла filen:='input'+num+'.txt' .

Сообщение пользователю о выбранном файле.

#### Считывание данных

Открывается файл "filen" для считывания данных.

"max:=0".

Каждой ячейке массива связей присваивается “ложь”

Пока файл не кончился считываются пара стран, в массив связей с индексами: [страна с меньшим номером, с большим] присваивается значение истина.

Закрывается файл "filen".

Блок, определяющий минимальное количество цветов.

Начальное количество цветов = 1.

Повторять действия:

Повысить количество цветов на единицу.

Пока не возможна раскраска всех стран данным количеством цветов.

**Запись данных.**

Создать файл "Output.txt". Считать в него количество цветов. Считать в него список раскраски стран. Закрыть файл.

#### Текст программы

program mag;

uses

crt;

var

num:char;

filen:string;

g:array [1..100]of integer;

s:array[1..100,1..100]of boolean;

max,s1,s2,j,n,i,a:integer;

f:file of char;

f1:text;

{ Функция считывает текущее число,

из файла связанного с пеpеменной - f. }

function get:integer;

var

k,c:integer;

s:string;

ch:char;

begin

s:=''; k:=50;

while ((s='')or(k=43)or(k=45)or((k>47)and(k<58)))and(not eof(f)) do

begin

read(f,ch); k:=ord(ch);

if (k=43)or(k=45)or((k>47)and(k<58)) then s:=s+ch;

end;

val(s,c,k);

if max< c then max:=c;

get:=c;

end;

{ Функция проверяет, можно ли страну - i

закрасить цветом - g[i]. }

function pr:boolean;

var j:integer;

begin

pr:=true;

if i>1 then

for j:=1 to i-1 do

if s[j,i] then

if g[i]=g[j] then pr:=false;

end;

function gen(n:integer):boolean;

begin

for j:=1 to max do

g[j]:=0;

i:=1;

repeat

repeat

g[i]:=g[i]+1;

until pr or (g[i]>n);

if (g[i]>n) then

begin

g[i]:=0; i:=i-1;

end

else i:=i+1;

until (i=1)or(i>max);

gen:=i>max;

end;

begin

clrscr;

writeln(' Haжмите цифру, указывающую номер файла');

writeln(' с которого будут счтываться данные. ');

writeln('');

for i:=1 to 3 do

writeln(' ',i,' - файл input',i,'.txt');

{Запpос номеpа файла.}

repeat

num:=readkey;

if not((num='1')or(num='2')or(num='3')) then

writeln(' Вы в чем-то ошиблись');

until (num='1')or(num='2')or(num='3');

filen:='input'+num+'.txt';

writeln('');

writeln(' Выбран файл - ',filen);

writeln('');

{Считывание данных}

Assign(f,filen); Reset(f);

max:=0;

for s1:=1 to 100 do

for s2:=1 to 100 do

s[s1,s2]:=false;

while not eof(f) do

begin

s1:=get; s2:=get;

if s1>s2 then s[s2,s1]:=true else s[s1,s2]:=true;

end;

Close(f);

n:=1;

repeat

n:=n+1;

until gen(n);

{ Вывод данных в файл - 'output.txt'.}

Assign(f1,'output.txt'); Rewrite(f1);

writeln(f1,' Число стран = ',max);

writeln(f1,'');

writeln(f1,'N cтран = ',n);

for j:=1 to max do

writeln(f1,'cтрана - ',j,', цвет - ',g[j],' ');

Close(f1);

end.

Koнец.

**Данные из Input1.txt:**

1 2 1 7 2 7 2 3 2 8 3 4 3 9 3 8 4 9 4 5 5 6 5 9 5 13

6 13 6 14 6 15 7 8 7 11 8 9 8 10 8 11 9 10 9 12 9 13

10 11 10 12 10 14 11 15 12 14 13 14 14 15

**Результат в output.txt:**

Число стран = 15

N стран = 4

страна - 1, цвет - 1

страна - 2, цвет - 2

страна - 3, цвет - 1

страна - 4, цвет - 2

страна - 5, цвет - 1

страна - 6, цвет - 2

страна - 7, цвет - 3

страна - 8, цвет - 4

страна - 9, цвет - 3

страна - 10, цвет - 1

страна - 11, цвет - 2

страна - 12, цвет - 2

страна - 13, цвет - 4

страна - 14, цвет - 3

страна - 15, цвет - 1

**1.** Дана функция  **f(x) = a+b\*x\*Sin(c\*x+d),** в которой коэффициенты **a, b, c, d** определяются функцией **Rnd** с использованием процедуры **Randomize**. Постоянная **k** определяется в интервале **[0.1;0.2]** c шагом **0.01**. Составить программу определения первой точки пересечения прямой **y = k\*x** с функцией  **f** для **x>0.**

**Текст программы:**

CLS

RANDOMIZE 1000

INPUT "точность"; t

a1 = RND(100)

q1 = RND(90)

q2 = RND(80)

q3 = RND(110)

DEF fnf (x) = a1 + q1 \* x \* SIN(q2 \* x + q3) - k \* x

a = 0: b = 110

FOR k = .1 TO .2 STEP .01

GOSUB pod1

GOSUB pod2

PRINT "при k="; k

cor = (x1 + x2) / 2: y = fnf(cor) + k \* x

PRINT "точка пересечения"

PRINT "x="; cor; "y="; y;

a = x2

NEXT k

PRINT "Точность"; t

END

a = x2

NEXT k

PRINT "Точность"; t

END

pod1: 'отделение корней в интервале а,в'

s = 10 \* t: x = a

200 : p = fnf(x) \* f(x + s)

IF p > 0 THEN 1

x1 = x: x2 = x + s: RETURN

1 x = x + s: IF x <= b - s THEN 200

RETURN

pod2: 'уточнение корня'

10 x = (x1 + x2) / 2: y = fnf(x)

IF fnf(x1) \* y > 0 THEN x1 = x ELSE x2 = x

IF ABS(x2 - x1) > t THEN 10

RETURN

**Данные:**

a = 0

b = 110

точность? .001

**Результаты:**

при k= .1

точка пересечения

x= .0096875 y= .7444314

при k= .11

точка пересечения

x= .0196875 y= .7470496

при k= .12

точка пересечения

x= .0296875 y= .7497169

при k= .13

точка пересечения

x= .0396875 y= .7524328

при k= .14

точка пересечения

x= .0496875 y= .7551972

при k= .15

точка пересечения

x= .0596875 y= .7580096

при k= .16

точка пересечения

x= 6.968751E-02 y= .7608697

при k= .17

точка пересечения

x= 7.968751E-02 y= .7637773

при k= .18

точка пересечения

x= .0896875 y= .7667319

при k= .19

точка пересечения

x= .0996875 y= .7697333

**2.** Известны координаты вершин четырехугольника **A, B, C, D**. Найти точку пересечения его диагоналей и наибольший радиус окружности, которая имеет центр в этой точке и полностью лежит внутри этого четырехугольника. Координаты точек расположить на гибком диске.

**Текст программы:**

REM Программа нахождения точки пересечения диагоналей 4-х угольника

REM и наибольшего радиуса окружности лежащей в нем.

CLS

OPEN "a:zad2.dat" FOR INPUT AS #1

OPEN "a:zad2.txt" FOR OUTPUT AS #2

DIM r(6)

INPUT #1, x1, y1, x2, y2, x3, y3, x4, y4

x = ((x3 \* y2 - x2 \* y3) \* (x4 - x1) - (x4 \* y1 - x1 \* y4) \* (x3 - x2)) / ((y4

- y1) \* (x3 - x2) - (y3 - y2) \* (x4 - x1))

y = ((y4 - y1) \* (x - x1) / (x4 - x1)) + y1

r(1) = ABS((x2 - x1) \* y - (y2 - y1) \* x + x1 \* y2 - y1 \* x2) / SQR((x2 - x1) ^

2 + (y2 - y1) ^ 2)

r(2) = ABS((x3 - x2) \* y - (y3 - y2) \* x + x2 \* y3 - y2 \* x3) / SQR((x3 - x2) ^

2 + (y3 - y2) ^ 2)

r(3) = ABS((x4 - x3) \* y - (y4 - y3) \* x + x3 \* y4 - y3 \* x4) / SQR((x4 - x3) ^

2 + (y4 - y3) ^ 2)

r(4) = ABS((x1 - x4) \* y - (y1 - y4) \* x + x4 \* y1 - y4 \* x1) / SQR((x1 - x4) ^

2 + (y1 - y4) ^ 2)

r(5) = ABS((x3 - x1) \* y - (y3 - y1) \* x + x1 \* y3 - y1 \* x3) / SQR((x3 - x1) ^

2 + (y3 - y1) ^ 2)

r(6) = ABS((x4 - x2) \* y - (y4 - y2) \* x + x2 \* y4 - y2 \* x4) / SQR((x4 - x2) ^

2 + (y4 - y2) ^ 2)

min = r(1)

FOR i = 2 TO 6

IF r(i) = 0 THEN GOTO 5

IF r(i) < min THEN min = r(i)

5 NEXT i

PRINT "Точка пересечения диагоналей O ("; x; ","; y; ")"

PRINT "Радиус окружности лежащей в 4-х угольнике ABCD, R=", min

PRINT #2, "Точка пересечения диагоналей O ("; x; ","; y; ")"

PRINT #2, "Радиус окружности лежащей в 4-х угольнике ABCD, R=", min

**Данные с диска а:**

1,1 2,1 1,2 2,2

**Результаты:**

Точка пересечения диагоналей O ( 1.5 , 1.5 )

Радиус окружности лежащей в 4-х угольнике ABCD, R= .5

**3.** Элементы двумерного массива **Q(m, n) (m>n)** определяются по одной из формул :

- если номер строки  **i** < номера столбца **j** ,то элемент **Q(i, j)** вычисляется по формуле

**- если i < j, то** **Q(i, j) = eRnd\*Sin(Rnd-0.5),**

- если  **i = j, то Q(i, j)=Log(10\*Rnd)/Sin(Rnd),**

**-** если **i > j, то Q(i, j)=Rnd\*Q(j,i).**

По полученным таким образом данным найти и выдать на печать :

- строку с наибольшим средним значением элементов,

- строку, в которой разность рядом стоящих элементов - наименьшая,

- Элементы квадратной матрицы **(n, n),** c максимальной суммой диагональных элементов.

**Текст программы:**

CLS

2 INPUT "Число строк столбцов"; m, n

IF m >= n THEN 1

PRINT "Число строк должно быть больше числа столбцов": GOTO 2

1 DIM q(m, m)

FOR i = 1 TO m

FOR j = 1 TO m

IF i < j THEN q(i, j) = EXP(RND) \* SIN(RND - .5)

IF i = j THEN q(i, j) = LOG(10 \* RND) / SIN(RND)

IF i > j THEN q(i, j) = RND \* q(j, i)

NEXT j

NEXT i

FOR i = 1 TO m

FOR j = 1 TO n

PRINT USING "##.### "; q(i, j);

NEXT j

PRINT

NEXT i

DIM s(m)

FOR i = 1 TO m

f = 0

FOR j = 1 TO n

f = f + q(i, j)

NEXT j: s(i) = f / n

NEXT i

PRINT

PRINT "Строка с наибольшим средним значением"

max = s(1)

FOR j = 1 TO n

FOR i = 1 TO m

IF s(i) >= max THEN max = s(i): ii = i

NEXT i

PRINT USING "##.####"; q(ii, j);

NEXT j

PRINT

PRINT

PRINT "Строка в которой разность стоящих рядом эл-тов наименьшая"

min = ABS(q(1, 1) - q(1, 2))

FOR i = 1 TO m

FOR j = 2 TO n

z = ABS(q(i, j - 1) - q(i, j))

NEXT j

IF z >= min THEN 33

min = z: kk = i

33 NEXT i

FOR i = 1 TO n

PRINT USING "##.#### "; q(kk, i);

NEXT i

PRINT

s = 0

FOR i = 1 TO n

s = s + q(i, i)

NEXT i

k = 0

FOR i = 1 TO n

k = k + q(i + 1, i)

NEXT i

k(i) = k

PRINT

PRINT "Эл-ты квадратной матрицы (n,n) с макс. суммой диагональных эл-тов"

max = s

FOR i = 1 TO n

FOR j = 1 TO n

IF k(i) >= s THEN PRINT USING " ##.#### "; q(i + 1, j);

IF k(i) < s THEN PRINT USING " ##.#### "; q(i, j);

NEXT j

PRINT

NEXT i

**Данные:**

Число строк столбцов? 4,3

**Результаты:**

Исходная матрица

3.842 -0.373 0.367

-0.304 43.204 0.537

0.353 0.468 -0.708

0.201 -0.015 0.021

Строка с наибольшим средним значением

-0.303743.2038 0.5367

Строка в которой разность стоящих рядом эл-тов наименьшая

0.2005 -0.0149 0.0212

Эл-ты квадратной матрицы (n,n) с макс. суммой диагональных эл-тов

3.8424 -0.3729 0.3669

-0.3037 43.2038 0.5367

0.3530 0.4677 -0.7079

**4.** Составить таблицу значений определённого интеграла

**b**

**∫ (x\*tg(x)/(1+x2))\*dx**  при изменении верхнего предела в интервале **[t1;t2]**

**a**

c шагом **s.**

Интегрировать методом трапеций с заданной точностью. На печать выдать переменный предел, результат. Печать организовать оператором **Print Using.**

**Текст программы:**

CLS

OPEN "a:z4.txt" FOR OUTPUT AS #1

DEF fna (x) = (x \* TAN(x)) / (1 + x ^ 2)

INPUT "Нижняя граница интеграла"; a

INPUT "Введите изменение верхнего предела интеграла, b"; t1, t2

INPUT "Введите шаг"; s

INPUT "Точность"; t

FOR b = t1 TO t2 STEP s

PRINT "Верхняя граница"; a; "Точность"; t

PRINT #1, "Верхняя граница"; a; "Точность"; t

GOSUB INT2

PRINT USING " b=####.##### INTEGR=####.#####"; b; IN1

PRINT #1, USING " b=####.##### INTEGR=####.#####"; b; IN1

NEXT b

END

INT1:

dx = (b - a) / (n - 1)

IN1 = (fna(a) + fna(b)) / 2

x = a

5 x = x + dx

IN1 = IN1 + fna(a + dx)

IF x < b THEN 5

IN1 = IN1 \* dx

RETURN

INT2:

n = 100

45 GOSUB INT1: R = IN1

n = n + 20: GOSUB INT1

IF ABS(R - IN1) > t THEN 45

RETURN

**Данные:**

Нижняя граница интеграла? 2

Введите изменение верхнего предела интеграла, b? 3,5

Введите шаг? .9

Точность? .001

**Результаты:**

Верхняя граница 2 Точность .001

b= 3.00000 INTEGR= -0.86889

Верхняя граница 2 Точность .001

b= 3.90000 INTEGR= -1.64549

Верхняя граница 2 Точность .001

b= 4.80000 INTEGR= -2.32331