**684**В данной действительной квадратной матрице порядка n найти наибольший по модулю элемент. Получить квадратную матрицу порядка *n-1*путем выбрасывания из исходной матрицы какой-нибудь строки и столбца, на пересечении которых расположен элемент G найденным значением.

Код:

**program** N684;

**const** t = 8;

**type**

//введение собственного типа матрицы

matrix=**array**[1..t,1..t] **of** real;

nmatrix=**array**[1..t-1,1..t-1] **of** real;

**var**

n,ii,jj:byte; //ограничивающия константа/МАКСимальные позиции

a:matrix; //сама матрица

b:nmatrix;

max:real; //максимальный элемент

Logic: boolean;

Question: Char;

Way: string;

YourFile: text;

//Все глобальные переменные выше - перелитаются с несколькими процедурами

//процедура ввода начальной матрицы

**procedure** input(**var** a:matrix; **var** n:byte);

**var**

i,j:byte; //позиции

**begin**

randomize;

//ограничиваем константу

**repeat**

write('Введите порядок матрицы N, не больше чем 8: ');

readln(n);

**until** n<9;

//заполнение случайными числами матрицу (от -99 до 99)

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

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

A[i,j]:=-100+random(200)+(random(10)/10);

**end**;

//поиск max по модулю

**procedure** maximum(**var** a:matrix; **var** max:real; **var** n,ii,jj:byte);

**var**

i,j:byte; //позиции

**begin**

max:=abs(A[1,1]);

ii:=1;

jj:=1;

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

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

**if** abs(A[i,j])>max **then begin**

max:=abs(A[i,j]);

ii:=i; jj:=j;

**end**;

**end**;

//обрезание и создание новой матрицы

**procedure** cutting(**var** a:matrix; **var** b:nmatrix; n,ii,jj:byte);

**var**

i,j,ni,nj:byte;

**begin**

ni:=0;

nj:=1;

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

**if** (i<>ii) **then begin**

ni:=ni+1;

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

**if** (j<>jj) **then begin**

b[ni,nj]:=a[i,j];

nj:=nj+1;

**end**;

nj:=1;

**end**;

**end**;

//вывод ответа(обе матрицы)

**procedure** output(**var** a:matrix; **var** b:nmatrix; **var** max:real; **var** n,ii,jj:byte);

**var**

i,j:byte;

**begin**

//вывод начальной матрицы

writeln('Исходная матрица:');

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

**for** j:=1 **to** n **do** write(A[i,j]:6:1);

writeln;

**end**;

writeln;

writeln('Максимальный по модулю элемент = ', A[ii, jj]:6:1);

//вывод уменьшенной матрицы

**for** i:=1 **to** n-1 **do begin**

**for** j:=1 **to** n-1 **do**

write(b[i,j]:6:1);

writeln;

**end**;

**end**;

//вывод в файл полной матрицы

**procedure** outputinfileCH1(**var** a:matrix; **var** YourFile:text);

**var**

l,k:byte;

**begin**

writeln(YourFile,'Исходная матрица: ');

**for** l:=1 **to** n **do begin**

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

write(Yourfile,a[l,k]:6:1);

**end**;

writeln(Yourfile);

**end**;

**end**;

//вывод в файл урезанной матрицы

**procedure** outputinfileCH2(**var** b:nmatrix; **var** YourFile:text; **var** ii:byte; **var** jj:byte);

**var**

l,k:byte;

**begin**

writeln(YourFile,'Максимальный по модулю элемент = ', A[ii, jj]:6:1);

writeln(Yourfile);

writeln(YourFile,'Уменьшенная матрица: ');

**for** l:=1 **to** n-1 **do begin**

**for** k:=1 **to** n-1 **do begin**

write(Yourfile,b[l,k]:6:1);

**end**;

writeln(Yourfile);

**end**;

writeln(Yourfile);

writeln(Yourfile);

**end**;

**begin**

Logic:=false; //следующие 6 строчек кода связаны с выводом в файл

writeln('Напиши адрес для создания отчета:');

Read(way);

Assign(YourFile, Way);

**if** FileExists(Way)= False **then** Rewrite(YourFile) **else** Append(YourFile);

**while** logic = false **do begin**

input(a,n); //ввод массива

outputinfileCH1(a,YourFile); //вывод в файл полной матрицы

maximum(a,max,n,ii,jj); //поиск максимального элемента матрицы по позиции

cutting(a,b,n,ii,jj); //обрезка матрицы

output(a,b,max,n,ii,jj); //вывод матрицы

outputinfileCH2(b,YourFile,ii,jj);

Writeln('Хотите использовать программу еще раз? Y\N'); //многократное использование программы

Read(Question);

**If** question = 'N' **then** Logic:= True;

**end**;

closefile(YourFile);

**end**.

