**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ**

**ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО** »

**Інститут прикладного системного аналізу**

**Кафедра математичних методів системного аналізу**

**Звіт**

**про виконання лабораторної роботи №4**

**з дисципліни «Алгоритмізація та програмування»**

Виконав:

студент I курсу, группи КА-07

Лещинський Богдан Денисович

Прийняв:

к.т.н., доцент Безносик О. Ю.

Київ – 2020

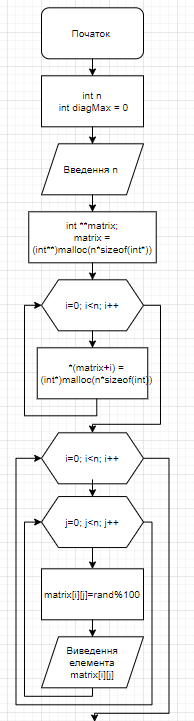
Завдання

Задана квадратна матриця. Виключити з неї рядок і стовпець, на перетині яких розташований максимальний елемент головної діагоналі.

Аналіз умови задачі

Виходячи з умови задачі можна сказати, що алгоритм задачі буде нелінійним, оскільки буде використовуватися динамічний двовимірний масив matrix для обробки даних. Для його створення потрібно використати можливості бібліотеки С stdlib.h , а для заповнення випадковими числами – бібліотеки stdlib.h і time.h. Окрім цього, можна сказати, що алгоритм буде циклічним з розгалуженнями, оскільки буде використовуватися вкладені цикли for для перебору значень масиву, і оператор if для визначення максимального елемента головної діагоналі і визначення його положення у матриці. Вхідні дані: n - порядок матриці, випадкові елементи matrix . Вихідні дані: елементи масиву matrix. Тип усіх даних – цілі числа int

Блок-схема алгоритму представлена на рисунку 1:



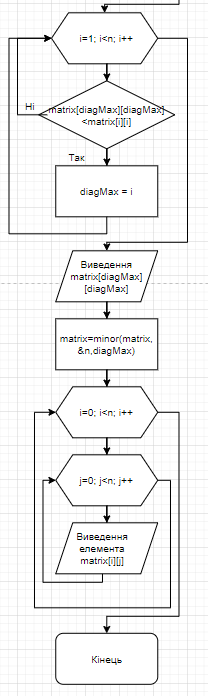
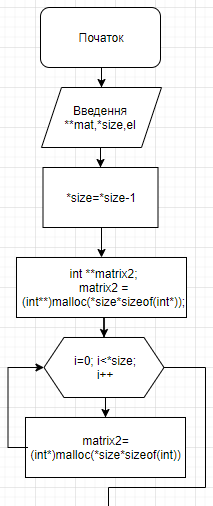


Рис.1

Блок-схема функції minor() представлена на рисунку 2:



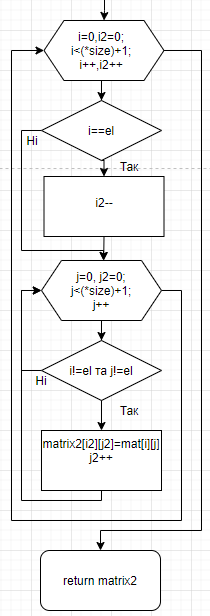


Рис.2

Код программи:

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include<time.h>

int\*\* minor(int\*\*mat,int\*size,int el){

\*size=\*size-1;

int \*\*matrix2;

matrix2 = (int\*\*)malloc(\*size\*sizeof(int\*));

for(int i=0; i<\*size; i++){

\*(matrix2+i) = (int\*)malloc(\*size\*sizeof(int));

}

for(int i=0,i2=0; i<(\*size)+1; i++,i2++){

if(i==el){

i2--;

}

for(int j=0, j2=0; j<(\*size)+1; j++){

if(i!=el && j!=el){

\*(\*(matrix2+i2)+j2) = \*(\*(mat+i)+j);

j2++;

}

}

}

for(int i=0; i<(\*size)+1; i++){

free(\*(mat+i));

}

free(mat);

return matrix2;

}

int main(){

srand(time(0));

int n;

int diagMax = 0;

printf("Enter natural n for matrix order: ");

scanf("%d",&n);

printf("\n");

int \*\*matrix;

matrix = (int\*\*)malloc(n\*sizeof(int\*));

for(int i=0; i<n; i++){

\*(matrix+i) = (int\*)malloc(n\*sizeof(int));

}

printf("The initial randomly filled matrix is: \n");

for(int i=0; i<n; i++){

for(int j=0; j<n; j++){

\*(\*(matrix+i)+j) = rand()%100;

printf("%d \t",\*(\*(matrix+i)+j));

}

printf("\n");

}

for(int i=1;i<n; i++){

if(\*(\*(matrix+diagMax)+diagMax)<\*(\*(matrix+i)+i)){

diagMax = i;

}

}

printf("The martix diagonal biggest element`s value is: %d\n The diagonal`s biggest element minor is: \n",\*(\*(matrix+diagMax)+diagMax));

matrix = minor(matrix,&n,diagMax);

for(int i=0; i<n; i++){

for(int j=0; j<n; j++){

printf("%d \t",\*(\*(matrix+i)+j));

}

printf("\n");

}

for(int i=0; i<n; i++){

free(\*(matrix+1));

}

free(matrix);

getchar();

return 0;

}

Результат роботи програми представлентй на рисунку 3:

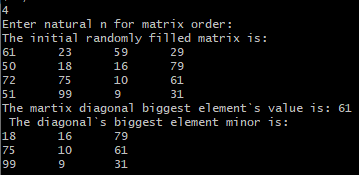


Рис.3

Висновки:

В ході виконання лабораторної роботи №4 було розроблено алгоритм розв’язку задачі по виключенню з матриці рядка та стовпця, на перетині яких розташований найбільший елемент головної діагоналі. В програмі використовувався двовимірний динамічний масив цілих чисел int matrix для збереження і обробки даних. Для створення динамічного масиву було використано можливості бібліотеки С stdlib.h, а для заповнення випадковими числами – бібліотеки stdlib.h і time.h. Алгоритм є циклічним з розгалуженнями, оскільки використовувався цикл for для перебору значень масиву, і оператор if для визначення максимального елемента головної діагоналі і визначення його положення у матриці. Усі дані: масив matrix, змінні n, diagMax - мають тип даних ціле число int. Для виведення використовується форматоване виведення.

.