**Міністерство Освіти І НАУКИ України**

**Національний університет "Львівська політехніка"**

Інститут **КНІТ**

Кафедра **ПЗ**

### ЗВІТ

До лабораторної роботи № 9

**На тему:** *“Динамічне виділення пам’яті”*

**З дисципліни:** *“Основи програмування”*

**Лектор:**

ст.викл. каф. ПЗ

Муха Т.О.

**Виконав:**

ст. гр. ПЗ-14

Губик А. С.

**Прийняв:**

ст.викл. каф ПЗ

Кутельмах Р. К.

« \_\_\_\_ » \_\_\_\_\_\_\_\_ 2022 р.

∑= \_\_\_\_ .

Львів – 2022

**Тема роботи:** динамічне виділення пам’яті.

**Мета роботи:** навчитися використовувати динамічну пам’ять, виділяти та звільняти її

засобами мови С.

**Індивідуальне завдання**

Ввести розмір квадратної матриці і її елементи. Елементи матриці розташувати в динамічній

пам'яті. Визначити номер стовпця, у якого сума елементів, розташованих нижче головної

діагоналі найменша. Обнулити елементи, використовувані при підрахунку цих сум.

**Теоретичні відомості**

2.1 Динамічні змінні

Бувають ситуації, коли заздалегідь невідомо, скільки даних потрібно обробити у

програмі. Особливо це стосується роботи з масивами – при оголошенні статичного масиву для

нього виділяється жорстко фіксована пам’ять. Однак, подекуди ця жорстко фіксована кількість

елементів масиву може перевищувати реальну кількість даних або ж навпаки – бути

недостатньою. Якщо масив не може вмістити потрібну кількість даних, доведеться

переписувати і повторно компілювати програму. Один з типових підходів при використанні

статичних масивів – виділити пам’ять з запасом, а заповнити даними лише частину масиву і в

якійсь змінній зберігати фактичну кількість елементів масиву. У цьому випадку виділена

пам’ять використовується неефективно. Крім того, статичні змінні “живуть” визначений час –

або до кінця роботи програми, або до кінця її певного блока. Взагалі кажучи, часто статична

змінна вже не є потрібною, а все ще займає пам’ять.

У мові С існує можливість динамічно розподіляти пам’ять у ході виконання програми,

тобто, виділяти стільки пам’яті, скільки її насправді потрібно, і звільнити її тоді, коли вона вже

не буде потрібною. Пам’ять виділяється з так званої вільної пам’яті. Розмір вільної пам’яті

залежить від операційної системи та моделі пам’яті комп’ютера. Динамічні змінні не мають

імен, а доступ до них здійснюється через вказівники. Динамічні об’єкти розміщаються у

“кучі” (англомовний термін-відповідник: heap). Для управління вільною пам’яттю у С

використовується набір функцій malloc(), calloc(), realloc() та free().

**Код програми**

Назва файлу: **matrix.c**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

void PrintMatrix(int \*\*m, int n)

{

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

if(i != 0)

printf("\n");

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

printf("%d ", m[i][j]);

}

}

void ReadMatrix(int \*\*m, int n)

{

char row[100] = "";

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

int j = 0;

fgets(row, 100, stdin);

char \*token = strtok(row, " \n");

while(token){

m[i][j] = atoi(token);

j++;

token = strtok(NULL, " \n");

}

if(j != n){

printf("There should be %d elements in a row!\n", n);

exit(1);

}

}

}

void CalculateThing(int \*\*m, int size, int \*col, int \*sum)

{

int MinSum = (unsigned)(-1) / 2;

int MinCol = 0;

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

int s = 0;

for(int j = i + 1; j < size; j++){

s += m[j][i];

m[j][i] = 0;

}

if (s < MinSum){

MinSum = s;

MinCol = i;

}

}

\*col = MinCol;

\*sum = MinSum;

}

int main()

{

int n, col = 0, sum = 0;

printf("Enter a size of your square matrix: ");

scanf("%d", &n);

if (n < 1){

printf("There shoud be at least 1 row!\n");

return 0;

}

getchar();

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

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

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

printf("Enter your matrix row by row:\n");

ReadMatrix(matrix, n);

CalculateThing(matrix, n, &col, &sum);

if(n < 2){

printf("There's no elements under diagonal\n");

return 0;

}

printf("\n");

PrintMatrix(matrix, n);

printf("\n");

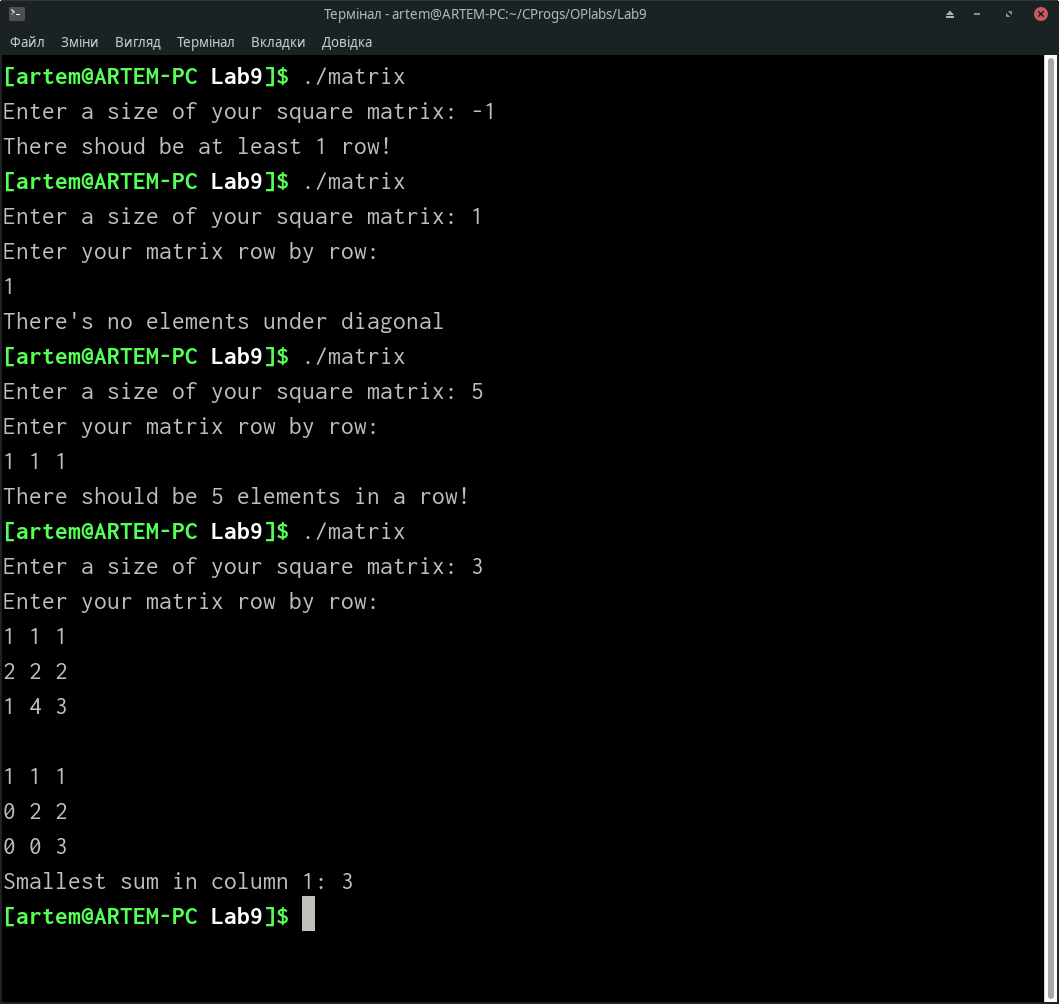
printf("Smallest sum in column %d: %d\n", col + 1, sum);

return 0;

}

**Протокол роботи**

**matrix.c**



**Висновки**

Динамічне виділення пам’яті стає корисним, коли наперед неможливо дізнатися скільки пам’яті знадобиться програмі.