# Лабораторна робота №10. Вступ до показчиків

## 1 ВИМОГИ

### 1.1 Розробник

* Хелемендик Дмитро Олегович;
* студент групи КІТ-121д;
* 29-груд-2021.

### 1.2 Загальне завдання

Дано масив масивів з N \* N цілих чисел. Елементи головної діагоналі записати в одновимірний масив, отриманий масив упорядкувати за зростанням.

## 2 ОПИС ПРОГРАМИ

### 2.1 Функціональне призначення:

Програма призначена для знаходження елементів головної діагоналі та запис їх в масив. Після цього елементу будуть упорядковані за зростанням. Програма працює за допомогою функцій, що задекларовані в *lib.h*, *stdlib.h* та *time.h*.

При заповненні двовимірного масиву та знаходженні головної діагоналі діють наступні *обмеження*:

- числа: від 0 до 100;

- розмір масивів залежить від константи N.

Результат зберігається у показчику *parray2*.

Демонстрація знайдених результатів передбачає виконання програми у відлагоднику.

### 2.2 Опис роботи основної функції:

Під час генерації чисел:

- для отримання псевдовипадкового числа (в діапазоні `[0..100]`) використовується наступна конструкція `rand() % 100`.

Опис розроблених структур і функцій наводиться на базі результатів роботи системи автодокументування *Doxygen*.

#### Функція заповнення двовимірного масиву

void push(int \*\*arr1, int n);

*Призначення*: заповнення двовимірного масиву псевдорандомними числами від 0 до 100.

*Опис роботи*: функція генерує випадкові числа та заносить їх у переданий показчик.

**Аргументи**:

- *\*\*arr1* - показчик двовимірного масиву;

- *n* - розмір масиву.

#### Функція для отримання діагоналі

void get\_diagonal(int \*\*arr1, int \*arr2, int n);

*Призначення*: отримує головну діагональ матриці.

*Опис роботи*: функція отримує головну діагональ показчика двовимірного масиву та записує її в переданий показчик.

**Аргументи**:

- *\*\*arr1* - показчик двовимірного масиву;

- *\*arr2* - показчик одновимірного масиву, в який записується діагональ;

- *n* - розмір масиву.

#### Основна функція

int main()

*Призначення*: головна функція.

*Опис роботи*:

- створення двовимірного масиву parray1 із `NxN` чисел за допомогою показчиків;

- створення масиву parray2 із `N` чисел за допомогою показчиків;

- генерація чисел для parray1 за допомогою функції `push`;

- отримання чисел, які знаходяться на діагоналі parray1, та запис їх в parray2 шляхом виклика функції `get\_diagonal`;

- cортування чисел в масиві parray2 за допомогою функції `qsort`;

- звільнення пам'яті;

- повертає функція код повернення з програми (0).

#### Структура проекту:

└── lab10

├── README.md

├── Doxyfile

├── Makefile

├── doc

│ ├── assets

│ ├── lab10.docx

│ ├── lab10.pdf

│ └── lab10.md

├── srс

│ ├── lib.c

│ ├── lib.h

│ └── main.c

└── test

└── test.c

#### 2.3 Важливі фрагменти програми

#### Початкові дані. Константи, перерахування

Розмір масиву

#define N 3

#### Генерація чисел

\*(\*(arr1 + i) + j) = rand() % 100;

#### Отримання головної діагоналі

\*(arr2 + i) = \*(\*(arr1 + i) + i);

#### 3. Варіанти використання

Для демонстрації результатів кожної задачі використовується:

- покрокове виконання програми в утиліті lldb;

**Варіант використання 1**: запуск програми у режимі відлагодження:

- запустити програму у відлагоднику lldb;

- поставити точку зупинки на функції main (строка 45);

- запустити програму;

- подивитись результати виконання програми, оглядаючи значення змінних parray1 та parray2.

Process 2745 stopped

\* thread #1, name = 'main.bin', stop reason = breakpoint 1.1

frame #0: 0x0000000000401320 main.bin`main at main.c:45:11

42 get\_diagonal(parray1, parray2, N); // записує діагоналі в parray2

43 qsort(parray2, N, sizeof(int), cmp); // відсортовує масив

44 // звільнення пам'яті

-> 45 for (int i = 0; i < N; i++) {

46 free(\*(parray1 + i));

47 }

48 free(parray1);

(lldb) p \*(int(\*)[3]) parray1[0]

(int [3]) $0 = ([0] = 59, [1] = 26, [2] = 15)

(lldb) p \*(int(\*)[3]) parray1[1]

(int [3]) $1 = ([0] = 38, [1] = 66, [2] = 50)

(lldb) p \*(int(\*)[3]) parray1[2]

(int [3]) $2 = ([0] = 70, [1] = 50, [2] = 63)

(lldb) p \*(int(\*)[3]) parray2

(int [3]) $3 = ([0] = 59, [1] = 63, [2] = 66)

## ВИСНОВКИ

При виконанні даної лабораторної роботи було набуто практичного досвіду в роботі з показчиками.