

# Лабораторна робота №11. Взаємодія з користувачем шляхом механізму введення/виведення

## 1 ВИМОГИ

### 1.1 Розробник

- Хелемендик Дмитро Олегович;
- студент групи КІТ-121д;
- 03-січ-2021.

### 1.2 Загальне завдання

Визначити зворотню матрицю. Робота повинна бути присвячена взаємодії з користувачем шляхом використання функцій `write()`, `read()`. При старті програми виводилась інформація об авторі, номері та темі лабораторної роботи.

## 2 ОПИС ПРОГРАМИ

### 2.1 Функціональне призначення:

Програма призначена для знаходження зворотної матриці. Програма працює за допомогою функцій, що задекларовані в *lib.h*, *stdlib.h*, *string.h*, *stdio.h* та *unistd.h*.

При запиті розміру матриці діють наступні обмеження:

- числа: від 2 до 4 включно;

При знаходженні оберненої матриці діють наступні обмеження:

- якщо визначник дорівнює нулю, то матриці не існує;

Результат зберігається у показчику *matrix\_res*.

Демонстрація знайдених результатів передбачає виконання програми у вікні консолі.

### 2.2 Опис логічної структури

Під час отримання даних з стандартного потоку:

- для отримання даних `stdin` використовується наступна конструкція:  
`fgets(buffer_for_stdin, sizeof(buffer_for_stdin), stdin);`

Під час друку матриці:

- для заповнення буферу даними матриці використовується наступна конструкція: `snprintf(buffer, sizeof(buffer), "%7.2f ", (*(mtx + r) + c));`

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

### **Функція отримання розміру матриці**

```
int size_of_matrix();
```

*Призначення:* отримання розміру матриці.

*Опис роботи:* запитує у користувача розмір матриці, який повинен бути від 2 до 4 включно.

### **Функція заповнення матриці**

```
void input_matrix(float **mtx, int size);
```

*Призначення:* заповнення матриці користувачем.

*Опис роботи:* функція запитує у користувача дані матриці та заповнює її ціма значеннями.

**Аргументи:**

- *mtx* - показчик на матрицю;
- *size* - розмір матриці.

### **Функція перетворення матриці на обернену**

```
int inverse_matrix(float **mtx, float **mtx_res, int size);
```

*Призначення:* отримання оберненої матриці.

*Опис роботи:* функція розраховує обернену матрицю та записує її в результуючу матрицю. Якщо визначник матриці дорівнює 0, то обернена матриця не існує.

### **Аргументи:**

- *mtx* - показчик на початкову матрицю;
- *mtx\_res* - показчик на обернену матрицю;
- *size* - розмір матриці.

### **Функція друку матриці**

```
void print_array(float **mtx, int size, int num);
```

*Призначення:* друк матриці.

*Опис роботи:* функція друкує матрицю.

### **Аргументи:**

- *mtx* - показчик на початкову матрицю;
- *size* - розмір матриці;
- *num* - номер матриці (1 - початкова, 2 - обернена).

### **Основна функція**

```
int main()
```

*Призначення:* головна функція.

*Опис роботи:*

- створюю буфер для даних *stdin* з розміром *N*;
- зчитую дані з *stdin* функцією *fgets()* та друкую їх за допомогою *write()*;
- створюю змінну розміру матриці та знаходю її функцією *size\_of\_matrix*;
- виділяю пам'ять для двох матриць (початкової та оберненої);
- записую дані в матрицю *matrix* за допомогою функції *input\_matrix*;
- далі перетворюю матрицю *matrix* на обернену функцією *inverse\_matrix*, яка записує обернену матрицю в *matrix\_res*;
- друкую початкову та обернену матриці шляхом виклику функції *print\_array* два рази;
- звільнюю пам'ять;

- успішний код повернення з програми (0).

### Структура проекту:

```
└─ lab11
    ├── assets
    │   └─ input.txt
    ├── doc
    │   ├── lab11.docx
    │   ├── lab11.md
    │   └─ lab11.pdf
    ├── Doxyfile
    ├── Makefile
    ├── README.md
    ├── src
    │   ├── lib.c
    │   ├── lib.h
    │   └─ main.c
    └─ test
        └─ test.c
```

## 2.3 Важливі фрагменти програми

### Початкові дані. Константи, перерахування

Розмір буферу

```
#define N 1000
```

### Отримання даних stdin використовується

```
fgets(buffer_for_stdin, sizeof(buffer_for_stdin), stdin);
```

## Друк матриці

```
for (int r = 0; r < size; r++) { // друкує матрицю
    write(1, " ", 2);
    for (int c = 0; c < size; c++) {
        snprintf(buffer, sizeof(buffer), "%7.2f ", (*(mtx + r) + c));
        write(1, &buffer, N);
    }
    write(1, "\n", 3);
}
```

### 3. Варіанти використання

Для демонстрації результатів кожної задачі використовується:

- виконання програми у вікні консолі.

**Варіант використання 1:** запуск програми у вікні консолі:

- запустити програму у консолі в папці проекту lab11 командою: `cat ./assets/input.txt | ./dist/main.bin`;
- ввести розмір матриці від 2 до 4 включно(наприклад 3);
- далі ввести дані матриці;
- подивитись результат програми.

```
dima@dima-VirtualBox:~/dev/programing-khelemendyk/lab11$ cat
./assets/input.txt | ./dist/main.bin
```

Виконав: Хелемендик Дмитро Олегович. Група: КІТ-121д. Лабораторна робота №11. Взаємодія з користувачем шляхом механізму введення/виведення.

Input the size of matrix NxN (max:4): 3

Input your matrix: 2

Input your matrix: 5

Input your matrix: 7

Input your matrix: 6

Input your matrix: 3

Input your matrix: 4

Input your matrix: 5

Input your matrix: -2

Input your matrix: -3

It's your start matrix:

( 2.00 5.00 7.00 )

( 6.00 3.00 4.00 )

( 5.00 -2.00 -3.00 )

It's your final matrix:

( 1.00 -1.00 1.00 )

( -38.00 41.00 -34.00 )

( 27.00 -29.00 24.00 )

### **ВИСНОВКИ**

При виконанні даної лабораторної роботи було набуто практичного досвіду у взаємодії з користувачем шляхом механізму введення/виведення.