# НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №*6*

із дисципліни «Програмування»

на тему

*«Динамічні масиви»*

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-63 | *асистент Громова В. В.* |
| *Артеменко Я.К.* |  |

Київ — 2017

# **ЗМІСТ**

[1 ПОСТАНОВКА ЗАДАЧІ 2](#_Toc478382182)

[1.1 Мета роботи 2](#_Toc478382183)

[1.2 Завдання на лабораторну роботу 2](#_Toc478382184)

[2 ОПИС ПРОГРАМИ 3](#_Toc478382185)

[ВИСНОВКИ 4](#_Toc478382186)

[Додаток А (блок-схема) 4](#_Toc478382187)

[Додаток Б (відповіді на контрольні запитання) 6](#_Toc478382188)

[Додаток В (текст програми) 7](#_Toc478382189)

# **1 ПОСТАНОВКА ЗАДАЧІ**

* 1. Мета роботи

Набути практичних навичок створення й опрацювання одновимірних і багатовимірних динамічних масивів.

* 1. Завдання на лабораторну роботу

Варіант №5

У матриці А розмірності m\*n знайти добуток від’ємних елементів у кожному стовпці та кількість нульових елементів у матриці.

**Постановка першого завдання**

Потрібно написати програму, у якій динамічні масиви створюються та опрацьовуються відповідно до завдання.

Програма повинна виконувати такі функції:

1. Уведення розміру вихідних матриці чи вектора.
2. Формування масиву (матриці) за допомогою бібліотекових функцій malloc() або calloc().
3. Заповнення масивів (можна випадковими числами).
4. Виконання завдання варіанту, формування нового масиву, результату.
5. Виведення на екран масиву й результату.
6. Видалення масивів за допомогою бібліотекової функції free().

# **ОПИС ПРОГРАМИ**

Дана програма містить 5 функцій: menu\_main, integer, valid\_bit, task, main.

Функція **menu\_main** виводить на екран дані про виконавця та надає можливість користувачу обрати завдання для виконання. Обрати завдання можна за допомогою стрілок «вверх» і «вниз» і за допомогою клавіші «Enter» перейти до його виконання. Пункти, які ми можемо обрати підсвічені білим кольором, а умови завдань виділяються сірим кольором. За допомогою оператора розгалуження «if – else» зроблено можливість переміщення «стрілки вибору» між пунктами меню та за допомогою десяткових індексів функціональних клавіш («80» – для стрілки вниз, а «72» – для стрілки вверх) зроблено можливим використання клавіш для вибору. За допомогою циклу «while» очікується натиснення клавіші Enter, після чого ця функція повертає значення вибраного пункту в функцію main.

Функції **integer та valid\_bit** перевіряють правильність введених даних. Якщо буде введена буква, а не цифра, то програма покаже повідомлення про помилку.

Функція **task** використовується для введення розмірності матриці та заповнення її псевдовипадковими числами. Після заповнення виведеться сама матриця та результат обчислення згідно умови завдання.

Функція **main** є головною і в ній викликаються всі функції і забезпечується робота меню за допомогу оператора «case».

# **ВИСНОВКИ**

На цій лабораторній роботі здобуто навички по роботі з динамічними масивами, вивчено методи заповнення масивів елементами та набуто практичні навички їх використання для написання програм.

# **Додаток А (блок-схема)**

Функція menu\_main:

Початок

System(“cls”);

Printf(“Інформація про виконавця”)

Printf(“Умова завдання”)

ні

If key = {0}

так

Cout<<”-> Продовжити”

Cout<<” Продовжити”

If key = {1}

так

Cout<<”-> Вихід”

Cout<<” Вихід”

Кінець

Функція task:

Початок

j=0,j<n,j++

i=0,i<n,i++

If (matr[i][j]==0)

ні

так

Count++

If (matr[i][j]<0)

ні

так

Count++

cout << setw(15) <<((mult==1)?0:mult);

cout <<"\n\nКоличество 0: "<< count;

Кінець

# **Додаток Б (відповіді на контрольні запитання)**

1. Статична пам’ять виділяється перед початком функціювання програми та вивільняється після завершення роботи програми. Динамічна пам’ять виділяється на вивільняється в процесі виконання програми.
2. Функції malloc, calloc, realloc і free.
3. Коли невідомо, скільки елементів даних буде опрацьовуватись.
4. За допомогою функції malloc:

int\* a = (int\*) malloc(10 \* sizeof(int));

або за допомогою функції calloc:

int\* a = (int\*) calloc(10, sizeof(int));

1. За допомогою функції free:

free(a);

1. Доступ до ділянок виділеної пам’яті виконується за допомогою операції індексування a[i].

# **Додаток В (текст програми)**

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#include <conio.h>

#include <windows.h>

#include <cstdlib>

#include <iostream>

#include <iomanip>

using namespace std;

int menu\_main()

{

HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

char key = 0;

int code;

do {

system("cls");

printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf(" \* Лабораторная работа №6 \*\n");

printf(" \* Динамические массивы \*\n");

printf(" \* Вариант №5 \*\n");

printf(" \* Группа KM-63 Артеменко Я.К.\*\n");

printf(" \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n");

key = (key + 2) % 2;

SetConsoleTextAttribute(hStdOut, (WORD)((0 << 4) | 8));

printf("ИНСТРУКЦИЯ!!! ДЛЯ НАВИГАЦИИ ПО МЕНЮ ИСПОЛЬЗУЙТЕ СТРЕЛКИ ВВЕРХ, ВНИЗ И КЛАВИШУ ENTER\n\n");

printf(" В матрице А размерности m\*n найти произведение отрицательных элементов в \n");

printf(" каждом столбике та количество нулевых элементов в матрице.\n\n");

SetConsoleTextAttribute(hStdOut, (WORD)((0 << 4) | 15));

if (key == 0) cout << " -> Заполнение массива с помощью генератора случайных чисел.\n" << endl;

else cout << " Заполнение массива с помощью генератора случайных чисел.\n" << endl;

if (key == 1) cout << " -> Выход." << endl;

else cout << " Выход." << endl;

code = \_getch();

if (code == 224)

{

code = \_getch();

if (code == 80) key++;

if (code == 72) key--;

}

} while (code != 13);

system("cls");

return key;

}

int integer() /\*валідація для цілих чисел\*/ {

char symbol[20];

int z = 0, k, res;

do

{

fflush(stdin); /\*занулення буфера від попередніх записів\*/

scanf("%s", &symbol);

while (symbol[z] >= '0' && symbol[z] <= '9')

z++;

res = atoi(symbol);

if (symbol[z])

printf("Введенный символ не целое число! Попробуйте еще раз: ");

else

return res;

} while (symbol[z]);

}

int valid\_bit(int a, int b) /\*валідація для бітових операцій на цілі числа (в залежності від потрібних значень)\*/ {

int c;

do

{

c = integer();

if (c < b || c > a)

printf("Слишком много символов! Попробуйте еще раз: ");

} while (c < b || c > a);

return c;

}

void task()

{

int n, m, top;

float\*\* matr;

cout << "Вводите только числа!" << endl;

cout << "\nВведите количество строк(1-10): ";

m = valid\_bit(10, 1);

cout << "Введите количество столбиков(1-10): ";

n = valid\_bit(10,1);

cout << "Введите максимальный и минимальный предел функции random(1-100): ";

top = valid\_bit(100, 1);

matr = (float\*\*)calloc(m, sizeof(float\*));

if (!matr)

{

cout << "Недостаточно памяти";

system("pause");

exit(1);

}

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

{

matr[i] = (float\*)calloc(n, sizeof(float));

if (!matr[i])

{

cout << "Недостаточно памяти";

system("pause");

exit(1);

}

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

matr[i][j] = ((rand() % (20 \* top + 1)) - top \* 10) / 10.0;

}

cout << "\nПолученный массив :\n";

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

{

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

{

cout << setw(15) << matr[i][j];

}

cout << endl;

}

cout << endl;

int count = 0;

cout << "Результат умножения: \n";

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

{

float mult=1;

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

{

if (matr[i][j] == 0)

count++;

if (matr[i][j] < 0)

mult \*= matr[i][j];

}

cout << setw(15) << ((mult==1)?0:mult);

}

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

free(matr[i]);

cout <<"\n\nКоличество 0: "<< count << endl;

free(matr);

cout << endl;

system("pause");

}

int main()

{

while (1)

{

setlocale(LC\_ALL, "RUS");

int i1 = menu\_main() + 1;

switch (i1)

{

case 1: {task(); break; }

case 2: {return 0; }

}

}

}