НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №*2*

із дисципліни «Основи програмування»

на тему

«Одновимірні й двовимірні масиви»

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-73 | *асистент Громова В.В.* |
| *Садченко М.В.* |  |

Київ — 2018

ЗМІСТ

[ВСТУП 3](#_Toc507441970)

[1 ПОСТАНОВКА ЗАДАЧІ 3](#_Toc507441971)

[1.1 Мета роботи 3](#_Toc507441972)

[1.2 Варіант та завдання лабораторної роботи 3](#_Toc507441973)

[2 ОПИС ПРОГРАМИ 3](#_Toc507441974)

[2.1 Опис першої програми 3](#_Toc507441975)

[2.2 Опис другої програми 4](#_Toc507441976)

[2.3 Опис бібліотек та функцій 4](#_Toc507441977)

[ВИСНОВКИ 5](#_Toc507441978)

[ПЕРЕЛІК ПОСИЛАНЬ 5](#_Toc507441979)

[ДОДАТОК А (Відповіді на контрольні запитання) 5](#_Toc507441980)

[ДОДАТОК Б (Код програми) 6](#_Toc507441981)

[Перша програма 6](#_Toc507441982)

[Друга програма 8](#_Toc507441983)

[ДОДАТОК В (Результат роботи програми) 11](#_Toc507441984)

[Перша програма 11](#_Toc507441985)

[Друга програма 11](#_Toc507441986)

# ВСТУП

Даний звіт створено для детального ознайомлення з лабораторною роботою №2, її завданням, структурою, розробкою програмного коду мовою програмування C, у супроводі детального розбору поставлених вимог.

# 1 ПОСТАНОВКА ЗАДАЧІ

## 1.1 Мета роботи

Вивчити базові операції для роботи з одновимірними та двовимірними масивами.

## 1.2 Варіант та завдання лабораторної роботи

Було виконано роботу за варіантом №24. Були виконані наступні

завдання:

**Завдання №1**:

1. Сформувати новий масив з елементів заданого цілочисельного масиву А (n), значення яких зустрічаються в ньому неодноразово. У новому масиві все значення повинні бути унікальними.
2. Впорядкувати масив комбінованим методом швидкого сортування з методом «бульбашки».

**Завдання №2**:

Переписати перші елементи кожного рядка матриці D (n, n), які більше 10, в масив В.

# 2 ОПИС ПРОГРАМИ

## 2.1 Опис першої програми

Спочатку на екрані з’являються варіанти натискання горячих клавіш. Користувачеві надається можливість обрати між запуском першої («1»), другої («2») підпрограм, очисткою консолі («8») та виходом («9»). Реалізовано за допомогою циклу do-while та switch-case.

Натиснувши «1», запуститься перша підпрограма, де потрібно ввести дані для заповнення масиву. Кожний ввід контролюється за допомогою валідації, що знаходиться в окремій функції та яка дозволяє пройти лише натуральному числу.

У другій підпрограмі за допомогою функції *rand()* всі заповнення виконуються машиною псевдовипадковими числами.

У кожній підпрограмі після заповнення масиву виконується відбір повторюваних чисел, скорочення масиву прибиранням копій та «бульбашкове» сортування. Сортування проводиться шляхом порівняння двох сусідів, більший з яких опиняється попереду. Після кожного етапу виводиться результат.

## 2.2 Опис другої програми

У другій програмі теж виводяться користувачеві варіанти доступу до двух підпрограм, очистки консолі та виходу з програми, відмінностей у реалізації немає.

При виборі клавіші «1», потрапляємо на початок першої підпрограми, де необхідно ввести кількість стовпців та рядків матриці (контролюється валідацією натуральних чисел). Далі вводяться елементи самої матриці (двовимірний масив), що контролюється валідацією дійсних чисел.

Друга підпрограма є протилежною до першої. Тут ввід всіх чисел, як і в першій програмі, теж виконується за допомогою псевдовипадкових чисел функціями *rand()* та *srand().*

Після заповнення матриці виконується пошук в кожному рядку першого числа, яке >=10. Далі всі ці числа виводяться на екран користувачеві у вигляді масиву.

## 2.3 Опис бібліотек та функцій

Для початку, опишемо використані бібліотеки у програмі:

<stdio.h> - відповідає за ввод\вивід даних;

<stdlib.h> - відповідає за функції, що займаються виділенням пам'яті, контроль процесу виконання програми, перетворення типів та інші;

<string.h> - містить функції для роботи з нуль-термінованими рядками і різними функціями роботи з пам'яттю;

<ctype.h> - містить оголошення функцій для класифікації символів;

<conio.h> - застосовується для створення текстового інтерфейсу;

<time.h> - містить типи та функції для роботи з датою та часом.

Опис функцій:

Програма починається з виконання функціїї int main(), яка визиває валідацію (у першій програмі – 2, у другій – 1 та 2):

1. float valid\_float()
2. int valid\_int()

# ВИСНОВКИ

Під час виконання лабораторної роботи були вивчені нюанси роботи з одномірними й двомірними масивами у мові програмування С. Також ми ознайомились з операціями вводу/виводу даних та з різними сортуваннями масивів.

# ПЕРЕЛІК ПОСИЛАНЬ

1. Програмування мовою С. Методичні вказівки до виконання лабораторних робіт із кредитного модуля «Програмування-2. С.» для студентів спеціальності 113 «Прикладна математика» [Текст] / В. В. Громова. — К. : НТУУ «КПІ», 2016. —5- 34 с.

# ДОДАТОК А (Відповіді на контрольні запитання)

1. *Які способи опису масивів застосовують у С?*

У загальному випадку масив — це набір елементів даних одного типу. Одновимірні масиви можна асоціювати з компонентами вектора, а двовимірні — компонентами матриці.

1. *Із чого починається індекс?*

Індексація масиву починається з нуля.

1. *Ініціалізація одновимірного масиву.*

int powers[4] = {1, 2, 4, 6};

1. *Звернення до елемента масиву за допомогою посилання.*

int q[5] = {1, 2, 3, 4, 5};

int\* r;

r = q;

int n; // елемент, до якого звертаємося

printf("%d\n", \*(r + n));

1. *Отримання адреси й значення елемента багатовимірного массиву.*

int array[3][3] = {{1, 2}, {2, 3}, {3, 4}};

printf(“%d”, array[1][1]); // надрукує 3;

int \*\*ptrArray = array[1][1]; // отримаємо адресу 3

1. *Посилання та масиви.*

Якщо pa — це посилання на ціле значення, описане як int\* pa, то присвоювання pa = &a[0] установлює в pa посилання на перший елемент a, тобто pa містить адресу a[0]. Присвоювання x = pa копіює вміст a[0] в x.

1. *Як отримати доступ до елемента багатовимірного масиву?*

int array[3][3] = {{1, 2}, {2, 3}, {3, 4}};

printf(“%d”, array[2][1]); // надрукує 4

# ДОДАТОК Б (Код програми)

## Перша програма

#include <stdio.h>

#include <string.h> // strlen(), strchr()

#include <stdlib.h> // atoi(), srand,rand(), exit(0)

#include <ctype.h> // isdigit()

#include <time.h> // time()

#include <conio.h> // system("cls"), getch()

int valid\_int(int arg) {

char str[65];

int num;

while (1) {

int smb = 0;

scanf("%s", str); *//input of value*

int len = strlen(str);

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

if (isdigit(str[i]) == 0) {

++smb;

**break**;

}

}

if (arg == 1) {

if (smb == 0) **break**;

else printf("Need natural number!: ");

}

else {

if ((smb == 0) || ((smb == 1) && (str[0] == '-'))) {

**break**;

}

else printf("Need natural number!: ");

}

}

num = atoi(str); *//convert str to int*

return num;

}

int main() {

int choice, nv = 100;

int pnt = 0, iter = 0, k = 0;

int \*size = &nv; *// вказівник для switch*

int array[\*size];

int new\_arr[\*size];

printf("**\n**Press '1' to form user array.**\n**"\

"Press '2' to form random array.**\n**"\

"Press '8' to clear console.**\n**"\

"Press '9' to get out program.**\n**");

do {

choice = getch();

} while(!strchr("1289", choice));

switch (choice) {

case '1': {

printf("**\n**Enter number of elements: ");

\*size = valid\_int(1);

*// FILING in ARRAY*

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

printf("array[%d] = ", i);

array[i] = valid\_int(0);

}

pnt = 1;

**break**;

}

case '2': {

\*size = 3 + rand() %20;

time\_t t;

*// Intializes random number generator*

srand((unsigned) time(&t));

printf("**\n**Enter max range: ");

int max\_ran = valid\_int(1);

*// FILING in ARRAY*

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

array[i] = -max\_ran/2 + rand() %(max\_ran+1);

printf("array[%d] = %d**\n**", i, array[i]);

}

pnt = 1;

**break**;

}

case '8': {

printf(" 7 ballov :)**\n**");

pnt = 0;

system("cls");

**break**;

}

case '9': exit(0); **break**;

default: **break**;

}

if (pnt == 1) {

*// ВІДБІР ПОВТОРЮВАНИХ*

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

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

if (array[i] == array[j]) {

new\_arr[iter] = array[j];

++iter;

**break**;

}

}

}

*// МАСИВ ПІСЛЯ ВІДБОРУ ПОВТОРЮВАНИХ*

printf("initial array = {");

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

if (i < iter-1) {

printf("%d, ", new\_arr[i]);

}

else {

printf("%d", new\_arr[i]);

}

}

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

*// СКОРОЧЕННЯ МАСИВУ*

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

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

if (new\_arr[i] == new\_arr[j]) {

for (k = j; k < iter; ++k) {

new\_arr[k] = new\_arr[k+1];

}

--iter;

}

else ++j;

}

}

*// МАСИВ ПІСЛЯ СКОРОЧЕННЯ*

printf("middle array = {");

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

if (i < iter-1) {

printf("%d, ", new\_arr[i]);

}

else {

printf("%d", new\_arr[i]);

}

}

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

*// "БУЛЬБАШКОВЕ" СОРТУВАННЯ*

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

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

if (new\_arr[i-1] > new\_arr[i]) {

int switcher = new\_arr[i-1];

new\_arr[i-1] = new\_arr[i];

new\_arr[i] = switcher;

}

}

}

*// КІНЦЕВИЙ МАСИВ ПІСЛЯ СОРТУВАННЯ*

printf("final array = {");

for (int l = 0; l < iter; ++l) {

if (l < iter-1) {

printf("%d, ", new\_arr[l]);

}

else {

printf("%d", new\_arr[l]);

}

}

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

}

return main();

}

## Друга програма

#include <stdio.h>

#include <string.h> // strlen(), strchr()

#include <stdlib.h> // atoi(), atof(), srand,rand(), exit(0)

#include <ctype.h> // isdigit()

#include <time.h> // time()

#include <conio.h> // system("cls"), getch()

int valid\_int() {

char str[65];

int num;

while (1) {

int smb = 0;

scanf("%s", str); *//input of value*

int len = strlen(str);

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

if (isdigit(str[i]) == 0) {

++smb;

**break**;

}

}

if (smb == 0){

num = atoi(str); *//convert str to int*

if (num == 0) printf("not zero!: ");

else **break**;

}

else printf("Need natural number!: ");

}

return num;

}

float valid\_float() {

char str[65];

float num;

while (1) {

int smb = 0, dot = 0;

scanf("%s", str); *//input of value*

int len = strlen(str);

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

if (isdigit(str[i]) == 0) {

if (str[i] == '.') {

++dot;

}

else {

++smb;

**break**;

}

}

}

if ((dot < 2) && ((smb == 0) || ((smb == 1) && (str[0] == '-')))) {

**break**;

}

else printf("Need float number!: ");

}

num = atof(str); *//convert str to float*

return num;

}

int main() {

int choice, pnt = 0;

int nc = 100, nr = 100;

int \*col = &nc, \*row = &nr; *// вказівник для switch*

float matrix[\*row][\*col];

float new\_arr[\*row];

printf("**\n**Press '1' to form user matrix.**\n**"\

"Press '2' to form random matrix.**\n**"\

"Press '8' to clear console.**\n**"\

"Press '9' to get out program.**\n**");

do {

choice = getch();

} while(!strchr("1289", choice));

switch (choice) {

case '1': {

printf("**\n**Enter number of columns: ");

\*col = valid\_int();

printf("Enter number of rows: ");

\*row = valid\_int();

*// FILING in MATRIX*

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

for (int j = 0; j < \*col; ++j) {

printf("matrix[%d][%d] = ", i+1, j+1);

matrix[i][j] = valid\_float();

}

}

pnt = 1;

**break**;

}

case '2': {

\*col = 1 + rand() %10;

\*row = 1 + rand() %10;

time\_t t;

*// Intializes random number generator*

srand((unsigned) time(&t));

printf("**\n**Enter max range: ");

int max\_ran = valid\_int();

*// FILING in MATRIX*

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

for (int j = 0; j < \*col; ++j) {

matrix[i][j] = -max\_ran/2 + rand() %max\_ran;

printf("matrix[%d][%d] = %g**\n**", i+1, j+1, matrix[i][j]);

}

}

pnt = 1;

**break**;

}

case '8': {

printf(" 7 ballov :)**\n**");

pnt = 0;

system("cls");

**break**;

}

case '9': exit(0); **break**;

default: **break**;

}

if (pnt == 1) {

*// DEMONSTRATION of the MATRIX*

putchar('**\n**');

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

for (int j = 0; j < \*col; ++j) {

printf("%g ", matrix[i][j]);

}

putchar('**\n**');

}

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

new\_arr[i] = 0;

for (int j = 0; j < \*col; ++j) {

if (matrix[i][j] >= 10) {

new\_arr[i] = matrix[i][j];

**break**;

}

}

}

printf("**\n**array = {");

for (int k=0; k < \*row; ++k) {

if (k < \*row-1) {

printf("%g, ", new\_arr[k]);

}

else {

printf("%g", new\_arr[k]);

}

}

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

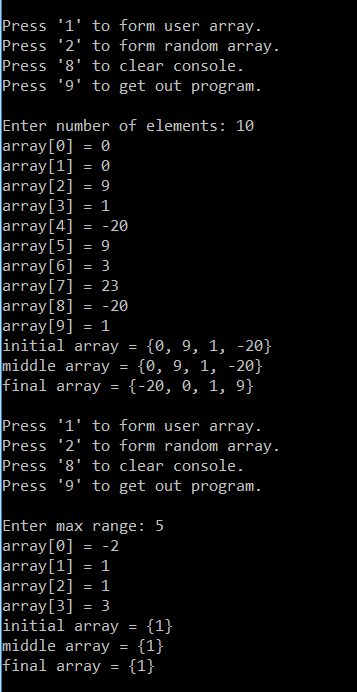
}

return main();

}

# ДОДАТОК В (Результат роботи програми)

## Перша програма



## Друга програма

