НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

до лабораторної роботи №4

із дисципліни «Програмування»

на тему

УКАЗІВНИКИ, ФУНКЦІЇ, РЕКУРСІЯ

|  |  |
| --- | --- |
| Виконав: | Керівник: |
| студент групи КМ-93 | *асистент Дрозденко О. М.* |
| *Пиндиківський Т. Р.* |  |

Київ — 2020

ЗМІСТ

[МЕТА РОБОТИ 3](#_Toc34108167)

[ПОСТАНОВКА ЗАДАЧІ 4](#_Toc34108168)

[ОСНОВНА ЧАСТИНА 5](#_Toc34108169)

[ВИСНОВКИ 9](#_Toc34108170)

[ДОДАТОК А 10](#_Toc34108171)

[ДОДАТОК Б 14](#_Toc34108172)

# МЕТА РОБОТИ

Ознайомитися з новим типом організації даних С (указівники) і набути практичних навичок його використання для написання програм; оволодіти синтаксисом написання функцій і їхніх прототипів, методикою складання і відлагодження програм, що містять функції, специфікою передачі параметрів у функцію та повернення одержаних результатів; вивчити методи використання алгоритмів і програм із рекурсією мовою С.

# ПОСТАНОВКА ЗАДАЧІ

***Завдання 1:***

1. Обчислити за допомогою функції кількість додатних, від’ємних і нульових елементів різних матриць.

***Завдання 2:***

1. Написати рекурсивну функцію знаходження цифрового кореня натурального числа. Цифровий корінь натурального числа отримують у такий спосіб. Якщо скласти всі цифри цього числа, потім все цифри знайденої суми і повторювати цей процес, то в результаті буде отримано однозначне число (цифра), яка і називається цифровим коренем даного числа.

# ОСНОВНА ЧАСТИНА

***Завдання 1 :***

Для виконання програми спочатку імпортуються наступні модулі :

* ***stdio.h*** – файл заголовку для стандартних операцій введення/виведення;
* ***time.h*** – бібліотека мови C, що містить функціонал для генерації довільних

чисел;

* ***stdlib.h*** – стандартна бібліотека мови С.
* ***isnumber.h*** – бібліотека, що містить функції для обробки даних, введених користувачем.

Змінні, що використовуються під час виконання завдання:

* ***rows*** (*int*) – ціла величина, в якій зберігається значення кількості рядків масиву;
* ***cols*** (*int*) –цілочисельна величина, що дорівнює кількості стовпців масиву;
* ***i, j*** (*int*) – допоміжні числа для ітерації по елементах масиву;
* ***choice*** (*int*) - ціле значення, що визначає режим заповнення масиву числами;
* ***array[][]*** *(float)* – масив чисел;
* ***prompt[]*** *(char) –* рядкова величина для підказки про введення даних користувачем.
* ***\*pointer*** *(float) –* вказівник на елементи масиву array
* ***counter\_positive, counter\_negative, counter\_zeros*** *(int) –* “лічильники” для підрахунку додатніх, від’ємних та нульвих значень масиву.

Хід виконання завдання:

1. Вводиться значення кількості рядків та стовпців масиву з використанням власної функції ***input\_int\_positive\_number(prompt)***, яка повертає введене користувачем ціле число (детальніше у додатку Б)*.*

***rows=input\_int\_positive\_number("\n\nEnter the number of rows in a 2d array");***

***cols=input\_int\_positive\_number("\n\nEnter the number of cols in a 2d array : ");***

1. Потім аналогічно вводиться з консолі значення змінної choice, що визначає метод заповнення масиву числами вручну або довільними числами (з використанням функції ***input\_int\_positive\_number***):

***choice=input\_int\_positive\_number("\n\nPress 1 - to fill array manually, 2 - to fill with random numbers : ");***

1. При виборі опції заповнення масиву довільними значеннями

«запускається» два цикли ***for()*** із подальшою ітерацією по значенню змінної ***i*** (від 0 до («кількість рядків»-1)) та ***j*** (від 0 до («кількість стовпців»-1)) та присвоєнню кожному елементу масиву ***array[i][j]*** довільного значення, що отримується при виклику функції ***rand(),*** яка повертає довільне число з вказаного діапазону.

1. При виборі опції заповнення масиву вручну «запускається» два цикли ***for()*** із подальшою ітерацією по значенню змінної ***i*** (від 0 до («кількість рядків»-1)) та ***j*** (від 0 до («кількість стовпців»-1)) та присвоєнню кожному елементу масиву ***array[i][j]*** значення, введеного користувачем із застосуванням функції ***input\_int\_positive\_number()*** (детальніше у додатку Б).
2. Сформувавши цикл ***for()*** із подальшою ітерацією по значенню вказівника ***\*pointer*** , що вказує на адресу кожного наступного елемента масиву, відбуваєтьсяя збільшення лічильників ***counter\_positive, counter\_negative, counter\_zeros*** на одиницю у випадку, коли значення вказівника додатнє/менше нуля/нуль.
3. В кінці з використанням функцій ***printf()*** із специфікаторами типів для цілих значень ***(“%d”),*** у консоль виводиться відповідна інформація про кількість додатних, від’ємних і нульових елементів масиву:

***printf("\n\nThe matrix contains %d positive numbers.", counter\_positive); printf("\n\nThe matrix contains %d negative numbers.", counter\_negative); printf("\n\nThe matrix contains %d zero numbers.", counter\_zeros);***

***Завдання 2 :***

Для виконання програми спочатку імпортуються наступні модулі :

• ***stdio.h*** – файл заголовку для стандартних операцій введення/виведення;

• ***time.h*** – бібліотека мови C, що містить функціонал для генерації довільних

чисел;

• ***stdlib.h*** – стандартна бібліотека мови С.

• ***isnumber.h*** – бібліотека, що містить функції для обробки даних, введених користувачем.

Змінні, що використовуються під час виконання завдання:

• ***rows*** (*int*) – ціла величина, в якій зберігається значення кількості рядків масиву;

• ***cols*** (*int*) –цілочисельна величина, що дорівнює кількості стовпців масиву;

• ***i*** (*int*) – допоміжне число для ітерації по елементах масиву;

• ***choice*** (*int*) - ціле значення, що визначає режим заповнення масиву числами;

• ***array[]*** (*float*) – масив чисел;

• ***prompt[]*** (*char*) – рядкова величина для підказки про введення даних користувачем.

* ***sum*** (*int*) – сума цифр числа;
* ***number*** (*int*) – ціле число, введене користувачем.
* ***digital\_root*** (*int*) – рекурсивна функція для обчислення

Хід виконання завдання:

1. Вводиться значення числа ***number*** з умови задачі з використанням власної функції ***input\_int\_positive\_number(prompt),*** яка повертає введене користувачем ціле число (детальніше у додатку Б).

***number=input\_int\_positive\_number("\n\nEnter a positive integer (natural) number : ");***

1. Потім, якщо число є меншим 10, то його значення виводиться у консоль з використанням функції ***printf():***

***printf("\n\nThe digital root of number %d is %d", number, digital\_root(number));***

3. У іншому випадку до значення змінної ***sum*** додається циклічно остача від ділення числа ***number*** на 10 та відбуається ділення ***number*** на 10, допоки значення змінної ***number*** є більшим за нуль. Ці команди рекурсивної функції ***digital\_root()*** викликаються допоки на вхід функція не отримає одноцифрове число, яке і є результатом виконання завдання та буде повернене функцією.

4. В кінці виводиться цифровий корінь числа ***number*** – результат виконання функції ***digital\_root()*** із застосуванням функції ***printf()*** та специфікатором типу ***“%d”.***

# ВИСНОВКИ

На цій лабораторній роботі було ознайомлено з новим типом організації даних С (указівники) і набуто практичні навички їх використання для написання програм; був вичений синтаксис для написання функцій і їхніх прототипів, методики складання і відлагодження програм, що містять функції, специфікою передачі параметрів у функцію та повернення одержаних результатів; вивчено методи використання алгоритмів і програм із рекурсією мовою С.

# ДОДАТОК А

1. ***Програмна реалізація задачі №1***

#include<stdio.h>

#include<time.h>

#include<string.h>

#include<stdlib.h>

#include<stdbool.h>

//#include"isnumber.h"

void program1 ()

{

// srand(time(0));

int rows, cols, i, j;

rows=input\_int\_positive\_number("\n\nEnter the number of rows in a 2d array : ");

cols=input\_int\_positive\_number("\n\nEnter the number of cols in a 2d array : ");

int choice;

do

{

choice=input\_int\_positive\_number("\n\nPress 1 - to fill array manually, 2 - to fill with random numbers : ");

} while(choice<0 || choice>2);

// char number\_chr[100];

float array[rows][cols];

for (i=0; i<rows; i++)

for (j=0; j<cols; j++)

array[i][j]=0;

int f, k;

// bool result=true;

char prompt[100];

//char array[10];

//sprintf(array, "%f", 3.123);

char data[100];

switch(choice)

{

case 1:

{

for (i=0; i<rows; i++)

{

for (j=0; j<cols; j++)

{

do

{

printf("\n\nEnter the value of number in %d row and %d column : ", i+1, j+1);

gets(data);

// if(!(is\_number(data)))

// printf("\nYou have entered wrong value");

} while((is\_number(data)!=true));

array[i][j]=atof(data);

}

}

printf("\n\nArray with manually entered numbers :\n");

for (i=0; i<rows; i++)

{

printf("\n");

for (j=0; j<cols; j++)

printf(" %.2f ", array[i][j]);

printf("\n");

}

break;

}

case 2:

{

printf("\n\nArray with random numbers :\n");

for (i=0; i<rows; i++)

{

printf("\n");

for (j=0; j<cols; j++)

{

array[i][j]= rand () % (rows\*cols) - (rand() % (rows\*cols)/2);

printf(" %.2f ", array[i][j]);

}

printf("\n");

}

break;

}

}

float \*pointer;

pointer=&array[0][0];

int counter\_negative=0, counter\_positive=0, counter\_zeros=0;

for(i=0;i<rows\*cols; pointer++, i++)

{

if(\*pointer<0)

counter\_negative++;

else if(\*pointer>0)

counter\_positive++;

else if((int)\*pointer==0)

counter\_zeros++;

}

printf("\n\nThe matrix contains %d positive numbers.", counter\_positive);

printf("\n\nThe matrix contains %d negative numbers.", counter\_negative);

printf("\n\nThe matrix contains %d zero numbers.", counter\_zeros);

}

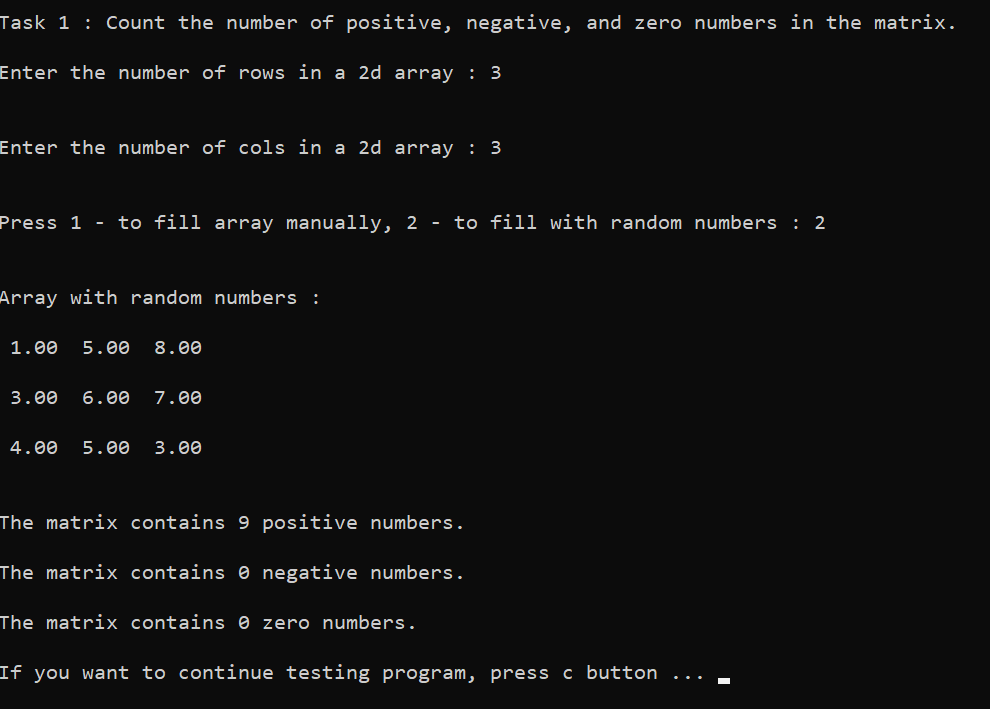
******

Рис.1 – Тестування завдання №1

1. ***Програмна реалізація задачі №2***

#include<stdio.h>

#include<time.h>

#include<string.h>

#include<stdlib.h>

//#include"isnumber.h"

long int sum\_ciphers(long int number)

{

long int sum=0, bufer=number;

while(number>0)

{

sum+=number%10;

number=(int)(number/10);

}

return sum;

}

long int digital\_root(long int number)

{

if(number>=0 && number<=9)

return number;

else

digital\_root(sum\_ciphers(number));

}

void program2()

{

long int number=0;

number=input\_int\_positive\_number("\n\nEnter a positive integer (natural) number : ");

printf("\n\nThe digital root of number %d is %d", number, digital\_root(number));

}

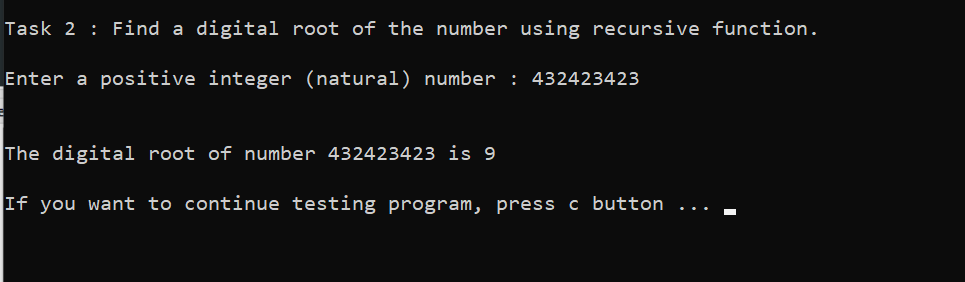
******

Рис.2 – Тестування завдання №2

# ДОДАТОК Б

Функція ***input\_int\_positive\_number (prompt)*** приймає від користувача введення даних цілого типу наступним чином : через функцію ***gets(),*** що приймає з консолі рядкову величину у тілі циклу ***while(),*** умовою виходу з якого є отримання рядка, що відповідає цілочисельній величині. Потім відбувається конвертація отриманих рядуових даних у число з використанням функції ***atoi()*** та повертається функцією ***input\_int\_positive\_number(prompt)*** як результат виконання.