**Лабораторна робота №4**

З дисципліни «Програмування»

Блочна структура програми. Функції

Виконав: Землянський Едуард

Група: КВ-22

**Постановка задачі**

Написати програму розв’язку завдання за варіантом з обов’язковим

використанням функцій для опису дій, які повинні повторюватися, а саме:

- введення елементів масиву;

- виведення елементів масиву;

- виконання над масивом дій, заданих за варіантом.

ПОКАЗАТИ ВИКОРИСТАННЯ ВИКЛИКУ ФУНКЦЫЙ ЯК

ОКРЕМИЙ ОПЕРАТОР і як частина виразу

**Вимоги до програми**

1. Алгоритми усіх функцій не повинні залежати від глобальних даних,

тобто масиви та інші дані, які передаються до функції та повертаються з

них, повинні передаватися як параметри або значення (результат) функції.

2. Використання видів параметрів у функціях повинно бути логічним з

точки зору швидкодії програми та економності використання оперативної

пам’яті комп’ютера.

3. В тестах значення елементів масивів підібрати таким чином, щоб в

алгоритмі обробки масиву були перевірені всі ситуації, які можуть

виникнути для заданого алгоритма, тобто продемонструвати повну

коректність та універсальність алгоритма.

**Варіант**

12. Дано три матриці цілих чисел A[7,4], B[6,5], C[6,8]. Обчислити суму

елементів окремо в кожному рядку заданих матриць.

**Текст програми:**

#include <stdio.h>

#include <stdlib.h>

#include <math.h>

#include <time.h>

void fill\_matrix(int rows, int columns, int matrix[rows][columns]){

    for (int i = 0; i <  rows; i++){

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

            matrix[i][j] = rand()%10;

        }

    }

}

void display\_matrix(int rows, int columns, int matrix[rows][columns]){

    for (int i = 0; i < rows; i++){

        printf("\n");

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

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

        }

    }

}

int calculate(int rows, int columns, int matrix[rows][columns]){

    int sum = 0;

    for (int i = 0; i <  rows; i++){

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

            sum += matrix[i][j];

        }

    }

    return sum;

}

int main(){

    srand(time(0));

    int A[7][4], B[6][5], C[6][8];

    // Çàïîâíåííÿ ìàñèâó À

    fill\_matrix(7, 4, A);

    // Âèâåäåííÿ ìàñèâó À

    printf("\nMatrix A:");

    display\_matrix(7, 4, A);

    // Çàïîâíåííÿ ìàñèâó B

    fill\_matrix(6, 5, B);

    // Âèâåäåííÿ ìàñèâó B

    printf("\n\nMatrix B:");

    display\_matrix(6, 5, B);

    // Çàïîâíåííÿ ìàñèâó C

    fill\_matrix(6, 8, C);

    // Âèâåäåííÿ ìàñèâó C

    printf("\n\nMatrix C: ");

    display\_matrix(6, 8, C);

    printf("\n");

    //Ñóìà åëåìåíò³â À

    int sum\_A = calculate(7, 4, A);

    printf("\nSum of A elements: %d", sum\_A);

    //Ñóìà åëåìåíò³â B

    int sum\_B = calculate(6, 5, B);

    printf("\nSum of B elements: %d", sum\_B);

    //Ñóìà åëåìåíò³â C

    int sum\_C = calculate(6, 8, C);

    printf("\nSum of C elements: %d", sum\_C);

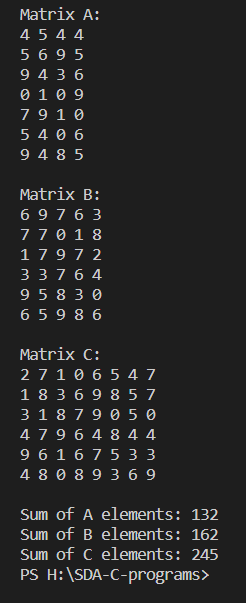
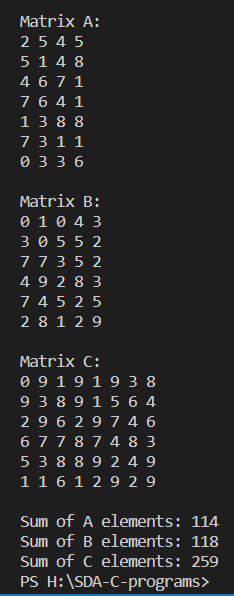
    return 0;

}

// 12. Äàíî òðè ìàòðèö³ ö³ëèõ ÷èñåë A[7,4], B[6,5], C[6,8]. Îá÷èñëèòè ñóìó

// åëåìåíò³â îêðåìî â êîæíîìó ðÿäêó çàäàíèõ ìàòðèöü.

**Тестування, результати для різних вхідних даних.**

**** **** 