**КПІ ім. Ігоря Сікорського**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

**Звіт до комп‘ютерного практикуму з курсу**

**“Основи програмування ”**

Прийняв Виконала

доцент кафедри ІПІ Студентка групи ІП-41

Лісовиченко О.І. Хижняк А. С.

“1” грудня 2024 р.

**Київ 2024**

**Комп‘ютерний практикум №6**

***Тема:*** *Використання динамічних  
масивів.*

**Завдання:**

Написати програму розв’язання системи лінійних алгебраїчних рівнянь (СЛАР) методом простої ітерації з використанням динамічних масивів.

**Текст програми:**

**#include** <stdio.h>

**#include** <stdlib.h>

**#include** <math.h>

**#define** max 10

**#define** min 2

**#define** min\_prec 1

**#define** max\_prec 15

**#define** min\_value -1000

**#define** max\_value 1000

int ValidInputI(int\* num);

int ValidInputD(double\* num);

void SimpleIter(int N, double Eps, double \*\*A, double \*B, double \*X, double \*XP);

int main()

{

    int n **=** 0;

    int precision;

    double \*\*a, \*b, \*x, \*xp;

    char end;

**do**{

    printf("\n\t\t\tThis program calculates SLAE\n\n");

**do**{

**do**{

            printf("Enter the number of rows and elements in rows(it must be equal) from %d to %d:", min, max);

        }**while**(**!**ValidInputI(&n) **||** n **<** min **||** n **>** max);

        a **=** (double\*\*)calloc(n, sizeof(double\*));

**for**(int i**=**0; i**<**n; i**++**)

        a[i] **=** (double\*)calloc(n, sizeof(double));

        b **=** (double\*)calloc(n, sizeof(double));

        x **=** (double\*)calloc(n, sizeof(double));

        xp**=**(double\*)calloc(n, sizeof(double));

**if** (a **==** NULL **||** b **==** NULL **||** x **==** NULL **||** xp **==** NULL)

        {

            printf("Failed to allocaate memory\n");

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

                free(a[i]);

            }

            free(a);

            free(b);

            free(x);

            free(xp);

        }

    }**while**(a**==**NULL **||** b **==** NULL **||** x **==** NULL **||** xp **==** NULL);

**do**{

        printf("Enter the number of decimal places from %d to %d:", min\_prec, max\_prec);

    } **while** (**!**ValidInputI(&precision)**||** precision **<** min\_prec **||** precision **>** max\_prec);

    printf("\n");

    double epsilon **=** pow(10, -precision);

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

        double temp;

**do**{

                temp**=**0;

                printf("Enter elements(a) of the %d row:\n\n", i+1);

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

**do**{

**do**{

                        printf("Enter %d element from %d to %d:",j+1, min\_value, max\_value);

                        }**while**(**!**ValidInputD(&a[i][j]) **||** a[i][j] **<** min\_value **||** a[i][j] **>** max\_value);

**if**(j**!=**i){

                            temp **+=**a[i][j];

                        }**else** **if**(j**==**i **&&** a[i][j]**==**0)

                        {

                            printf("Diagonal element cannot be zero. Please try again.\n\n");

                        }

                    }**while**(j**==**i **&&** a[i][j]**==**0);

                }

**if**(a[i][i]**<=**fabs(temp)){

                printf("The convergence condition is not met. Please try again.\n\n");

            }

        }**while**(a[i][i]**<=**fabs(temp));

**do**{

        printf("Enter b of %d row from %d to %d:", i+1, min\_value, max\_value);

        }**while**(**!**ValidInputD(&b[i]) **||** b[i]**<** min\_value **||** b[i] **>** max\_value);

        printf("\n");

    }

    SimpleIter(n, epsilon, a, b, x, xp);

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

        printf("x[%d] = %.\*lf\n", i+1, precision, x[i]);

    }

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

    free(a[i]);

    }

    free(a);

    free(b);

    free(x);

    free(xp);

    printf("Press Enter to continue or 0 to quit\n");

        end**=**getchar();

} **while** (end**!=**'0');

**return** 0;

}

int ValidInputI(int\* num)

{

    char check**=**0;

**if**(scanf("%d%c", num, &check) **!=**2 **||** check **!=** '\n')

    {

        printf("Error, not correct symbols!\n");

**while** (getchar()**!=**'\n');

**return** 0;

    }

**return** 1;

}

int ValidInputD(double\* num)

{

    char check**=**0;

**if**(scanf("%lf%c", num, &check) **!=**2 **||** check **!=** '\n')

    {

        printf("Error, not correct symbols!\n");

**while** (getchar()**!=**'\n');

**return** 0;

    }

**return** 1;

}

void SimpleIter(int N, double Eps, double \*\*A, double \*B, double \*X, double \*XP)

{

    double delta **=** 0;

    double max\_delta;

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

            XP[i] **=** B[i]/A[i][i];

        }

**do** {

    max\_delta **=** 0;

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

            double sum **=** 0;

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

**if** (j **!=** i) {

                    sum **+=** A[i][j] \* XP[j];

                }

            }

            X[i] **=** (B[i] - sum) / A[i][i];

            delta **=** fabs(X[i] - XP[i]);

**if** (delta **>** max\_delta) {

                max\_delta **=** delta;

            }

        }

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

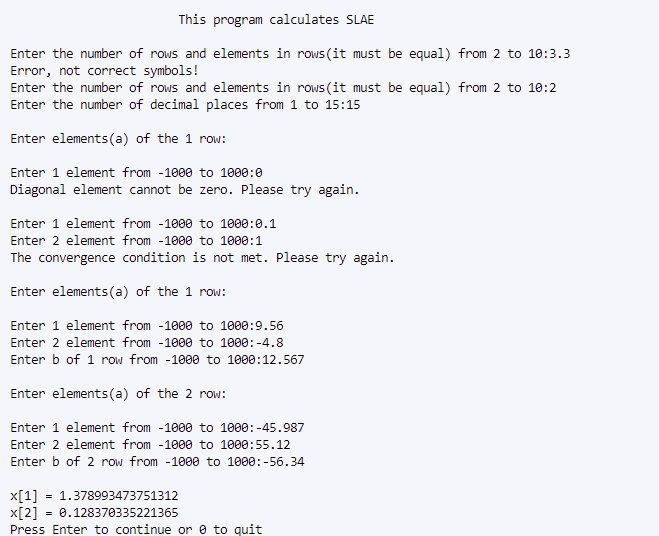
            XP[i] **=** X[i];

        }

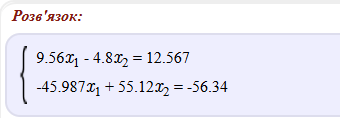
} **while** (max\_delta **>** Eps);

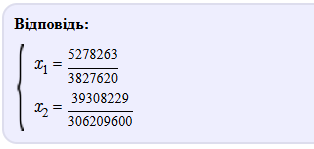
}

**Введені та одержані результати:**

****

**Теоретичні розрахунки:**

****

****

****

****

***Висновки:*** Теоретичні розрахунки відповідають отриманим. Програма працює

корректно. Програма вирішує поставлене завдання.