**КПІ ім. Ігоря Сікорського**

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформатики та програмної інженерії**

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

**“Основи програмування ”**

|  |  |
| --- | --- |
| Прийняв:  Асистент кафедри ІПІ  Пархоменко А.В.  11 січня 2023р. | Виконав:  Студент групи ІП-23  Зубарев М.К. |

**Київ 2023**

**Комп‘ютерний практикум №8**

**Тема:** Структури.

**Завдання:** Написати програму для обчислення комплексного опору заданого коливального контуру в залежності від частоти струму.

**Текст програми:**

#define M\_PI 3.1415

#include <stdio.h>

#include <conio.h>

#include <math.h>

#include <stdlib.h>

typedef struct point {

    double re;

    double lm;

}complex;

double F1,F2,dF,R1,R2,R,L,C;

double gF0(double L, double C) {

    double F0 = 1.0 / (2 \* M\_PI \* sqrt(L \* C));

    return F0;

}

void contur1() {

    printf("\nMethod #1\n");

    printf("\nEnter f min\n");

    scanf("%lf", &F1);

    printf("\nEnter f max\n");

    scanf("%lf", &F2);

    printf("\nEnter df\n");

    scanf("%lf", &dF);

    printf("\nEnter R\n");

    scanf("%lf", &R);

    printf("\nEnter L\n");

    scanf("%lf", &L);

    printf("\nEnter C\n");

    scanf("%lf", &C);

    double F = F1;

    double F0 = gF0(L, C);

    printf("\n F0 = %lf\n\n", F0);

    printf("\t\t\t~~~~Z~~~~\t\t\t\t~~~~F~~~~\n");

    do {

        complex p1;

        double gam = 2 \* M\_PI \* F;

        double a = L / C;

        double b = (-1 \* R) / (gam \* C);

        double c = R;

        double d = (gam \* L) - (1.0 / (gam \* C));

        p1.re = (a \* c + b \* d) / (pow(c, 2) + pow(d, 2));

        p1.lm = (b \* c - a \* d) / (pow(c, 2) + pow(d, 2));

        printf("\n\t\t%e + i \* %e\t\t%f", p1.re, p1.lm, F);

        F += dF;

    } while (F <= F2);

}

void contur2() {

    printf("\nMethod #2\n");

    printf("\nEnter f min\n");

    scanf("%lf", &F1);

    printf("\nEnter f max\n");

    scanf("%lf", &F2);

    printf("\nEnter df\n");

    scanf("%lf", &dF);

    printf("\nEnter R\n");

    scanf("%lf", &R);

    printf("\nEnter L\n");

    scanf("%lf", &L);

    printf("\nEnter C\n");

    scanf("%lf", &C);

    double F = F1;

    double F0 = gF0(L, C);

    printf("\n F0 = %lf\n\n", F0);

  printf("\t\t\t~~~~Z~~~~\t\t\t\t~~~~F~~~~\n");

    do {

        complex p1;

        double gam = 2 \* M\_PI \* F;

        double a = L / C;

        double b = (1 \* R) / (gam \* C);

        double c = R;

        double d = (gam \* L) - (1.0 / (gam \* C));

        p1.re = (a \* c + b \* d) / (pow(c, 2) + pow(d, 2));

        p1.lm = (b \* c - a \* d) / (pow(c, 2) + pow(d, 2));

        printf("\n\t\t%e + i \* %e\t\t%f", p1.re, p1.lm, F);

        F += dF;

    } while (F <= F2);

}

void contur3() {

    printf("\nMethod #3\n");

    printf("\nEnter f min\n");

    scanf("%lf", &F1);

    printf("\nEnter f max\n");

    scanf("%lf", &F2);

    printf("\nEnter df\n");

    scanf("%lf", &dF);

    printf("\nEnter R1\n");

    scanf("%lf", &R1);

    printf("\nEnter R2\n");

    scanf("%lf", &R2);

    printf("\nEnter L\n");

    scanf("%lf", &L);

    printf("\nEnter C\n");

    scanf("%lf", &C);

    double F = F1;

    double F0 = gF0(L, C);

    printf("\n F0 = %lf\n\n", F0);

    printf("\t\t\t~~~~Z~~~~\t\t\t\t~~~~F~~~~\n");

    do {

        complex p1;

        double gam = 2 \* M\_PI \* F;

        double a = R1 \* R2;

        double b = R1 \* ((gam \* L) - (1.0/(gam \*C)));

        double c = R1 + R2;

        double d = (gam \* L) - (1.0 / (gam \* C));

        p1.re = (a \* c + b \* d) / (pow(c, 2) + pow(d, 2));

        p1.lm = (b \* c - a \* d) / (pow(c, 2) + pow(d, 2));

        printf("\n\t\t%e + i \* %e\t\t%f", p1.re, p1.lm, F);

        F += dF;

    } while (F <= F2);

}

void contur4() {

    printf("\nMethod #4\n");

    printf("\nEnter f min\n");

    scanf("%lf", &F1);

    printf("\nEnter f max\n");

    scanf("%lf", &F2);

    printf("\nEnter df\n");

    scanf("%lf", &dF);

    printf("\nEnter R1\n");

    scanf("%lf", &R1);

    printf("\nEnter R2\n");

    scanf("%lf", &R2);

    printf("\nEnter L\n");

    scanf("%lf", &L);

    printf("\nEnter C\n");

    scanf("%lf", &C);

    double F = F1;

    double F0 = gF0(L, C);

    printf("\n F0 = %lf\n\n", F0);

    printf("\t\t\t~~~~Z~~~~\t\t\t\t~~~~F~~~~\n");

    do {

        complex p1;

        double gam = 2 \* M\_PI \* F;

        double a = (R1 \* R2) + (L / C);

        double b = (gam \* L \* R1) - (R2 / (gam \* C));

        double c = R1 + R2;

        double d = (gam \* L) - (1.0 / (gam \* C));

        p1.re = (a \* c + b \* d) / (pow(c, 2) + pow(d, 2));

        p1.lm = (b \* c - a \* d) / (pow(c, 2) + pow(d, 2));

        printf("\n\t\t%e + i \* %e\t\t%f", p1.re, p1.lm, F);

        F += dF;

    } while (F <= F2);

}

int main() {

    int choice;

    double res;

    char cont;

    printf("Choose circle(1-4)\n");

    scanf("%d", &choice);

    switch (choice) {

        case 1:

            contur1();

            break;

        case 2:

            contur2();

            break;

        case 3:

            contur3();

            break;

        case 4:

            contur4();

            break;

        }

    return 0;

}

**Введені та одержані результати:**

Зображення, що містить стіл

Автоматично згенерований опис

Зображення, що містить стіл

Автоматично згенерований опис

Зображення, що містить стіл

Автоматично згенерований опис

Зображення, що містить текст

Автоматично згенерований опис

**Зображення, що містить стіл

Автоматично згенерований описТеоретичні розрахунки:**

**Зображення, що містить стіл

Автоматично згенерований опис**

**Зображення, що містить стіл

Автоматично згенерований опис**

**Зображення, що містить стіл

Автоматично згенерований опис**

**Висновок:** теоретичні розрахунки відповідають отриманим із програми. Програма працює коректно й виконує поставлену задачу.