FAIZAN CHOUDHARY

20BCS021

PROGRAMMING LAB

13th December 2021

CODE: (code pasted in this format for readability)

```c
#include <stdio.h>
#include <stdbool.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

struct Complex
{
    double real;
    double imag;
};
struct Complex A, B, C;

void display (struct Complex C) {
    printf("%lf + %lfi\n", C.real, C.imag);
}

void add () {
    C.real = A.real + B.real;
    C.imag = A.imag + B.imag;
}

void subtract () {
    C.real = A.real - B.real;
    C.imag = A.imag - B.imag;
}

void multiply () {
    // z1 = p + iq and z2 = r + is
    // z1 * z2 = (pr - qs) + i(ps + qr)
    double p = A.real, q = A.imag, r = B.real, s = B.imag;
    C.real = (p * r) - (q * s);
    C.imag = (p * s) + (q * r);
}

void divide () {
    // z1 = a + ib and z2 = c + id
    // z1 / z2 = (ac + bd)/(c*c + d*d) + i(bc - ad)/(c*c + d*d)
    double a = A.real, b = A.imag, c = B.real, d = B.imag;
    C.real = (a*c + b*d) / (c*c + d*d);
```

```
    C.imag = (b*c - a*d) / (c*c + d*d);
}

double extract (char c[]) {
    int digits=1;
    bool flag=false;
    double number = 0;
    int num=0,i;
    for (i=0; i<strlen(c); i++) {
        if (c[i] == ' ')
            continue;
        if (c[i] == '.')
        // for decimal point numbers
            flag=true;
        else if (isdigit (c[i])) {
            // appending digits
            num = c[i] - '0';
            number = number * 10 + num;
            // if flag is true, it means there is some decimal point number (. has been
encountered)
            if (flag)
                digits *= 10;
        }
    }
    // updating number to have correct decimal place after division
    number = number / digits;
    // printf("%lf\n", number);
    return number;
}

void parse (char c[100]) {
    int i, k=0, counter=0;
    int neg = 1;
    double n;
    // to parse out numbers and pass it to extract() fxn
    char subs[50];
    for (i=0; i<strlen(c); i++) {
        if (c[i] == '-')
        // for negative numbers
            neg = -1;
        if (c[i] == 'i' || c[i] == ' ') {
            subs[k++] = '\0';
            n = extract(subs);
            k = 0;
            counter++;
            if (counter == 1) {
                A.real = (neg * n);
                neg = 1;
            }
            else if (counter == 2) {
                A.imag = (neg * n);
                neg = 1;
            }
            else if (counter == 3) {
```

```
                B.real = (neg * n);
                neg = 1;
            }
            else if (counter == 4) {
                B.imag = (neg * n);
                neg = 1;
            }
            // to skip ' + ' and ', '
            i+=2;
        }
        // copying number into a secondary string
        subs[k++] = c[i];
    }
}

int main() {
    char input[100];
    int i,ch;
    int flag = 0;
    printf("\nFAIZAN CHOUDHARY\n20BCS021\n");

    B:
    printf("\nEnter single string containing the two complex numbers (Ex: 0.123 + -9.0i, -
4.23 + 6.9i): ");
    if (flag == 1)
    getchar();
    scanf("%[^\n]*c", &input);

    parse(input);
        // printf ("%lf %lf %lf %lf", A.real, A.imag, B.real, B.imag);
    while (1) {
        flag = 1;
        printf("\nEntered complex numbers:\n");
        display(A);
        display(B);

        A:
        printf("\nMENU:\n1. Addition\n2. Subtraction\n3. Multiplication\n4. Division\n5.
Enter number again\n6. Exit\n");
        scanf("%d", &ch);

        switch (ch)
        {
            case 1: printf("\nAfter Addition: ");
                    add();
                    display(C);
                    break;
            case 2: printf("\nAfter Subtraction: ");
                    subtract();
                    display(C);
                    break;
            case 3: printf("\nAfter Multiplication: ");
                    multiply();
                    display(C);
```

```
                    break;
            case 4: printf("\nAfter Division: ");
                    divide();
                    display(C);
                    break;
            case 5: goto B;
            case 6: exit(0);
            default: printf("\nWrong choice! Enter again...\n");
                    goto A;
                    break;
        }
    }
    return 0;
}
```

OUTPUT:

```
FAIZAN CHOUDHARY
20BCS021

Enter single string containing the two complex numbers (Ex: 0.123 + -9.0i, -4.23 + 6.9i): -5 + 8.69i, 7.42 + -8.69i

Entered complex numbers:
-5.000000 + 8.690000i
7.420000 + -8.690000i

MENU:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Enter number again
6. Exit
1

After Addition: 2.420000 + 0.000000i
```

```
Entered complex numbers:
-5.000000 + 8.690000i
7.420000 + -8.690000i

MENU:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Enter number again
6. Exit
3

After Multiplication: 38.416100 + 107.929800i
```

```
Entered complex numbers:
-5.000000 + 8.690000i
7.420000 + -8.690000i

MENU:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Enter number again
6. Exit
5

Enter single string containing the two complex numbers (Ex: 0.123 + -9.0i, -4.23 + 6.9i): 0.123 + -9.0i, -4.23 + 6.9i
```

```
Entered complex numbers:
0.123000 + -9.000000i
-4.230000 + 6.900000i

MENU:
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Enter number again
6. Exit
2

After Subtraction: 4.353000 + -15.900000i
```