Computer Programming Lab

CEN-392

Program 12

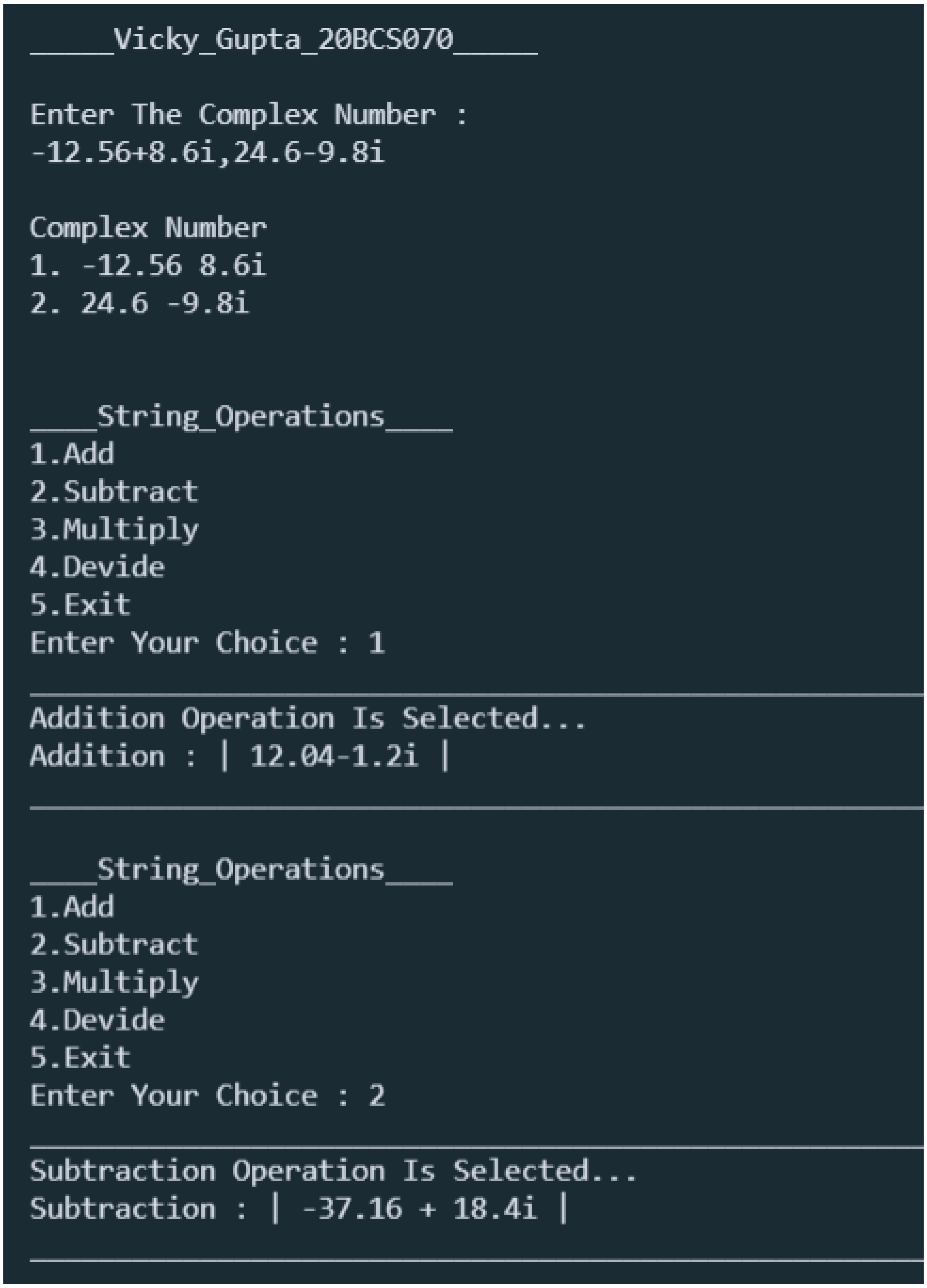
Code :-

```
#include <iostream>
using namespace std;

int lenght(char complex[])
{
    int len = 0;
    while (complex[len] != '\0')
        len++;
    return len;
}

void reverse(char str[])
{
    int len = lenght(str);
    for (int i = 0; i < len / 2; i++)
    {
        char ch = str[i];
        str[i] = str[len - 1 - i];
        str[len - 1 - i] = ch;
    }
}

bool check_decimal(char str[])
{
    int itr = 0;
    while (str[itr] != '\0')
    {
        if (str[itr++] == '.')
            return true;
    }
    return false;
}

void String_Integer(char str[], int s, int n, float arr[],
int indx)
{
    int ten_pow = 1;
    arr[indx] = 0;
    while (n >= s)
    {
        arr[indx] += ten_pow * (str[n--] - '0');
        ten_pow *= 10;
    }
}

void String_Decimal(char str[], int s, int n, float arr[],
int indx)
{
    float ten_pow = 0.1;
    while (s <= n)
    {
        arr[indx] += ten_pow * (str[s++] - '0');
        ten_pow /= 10;
    }
}
```

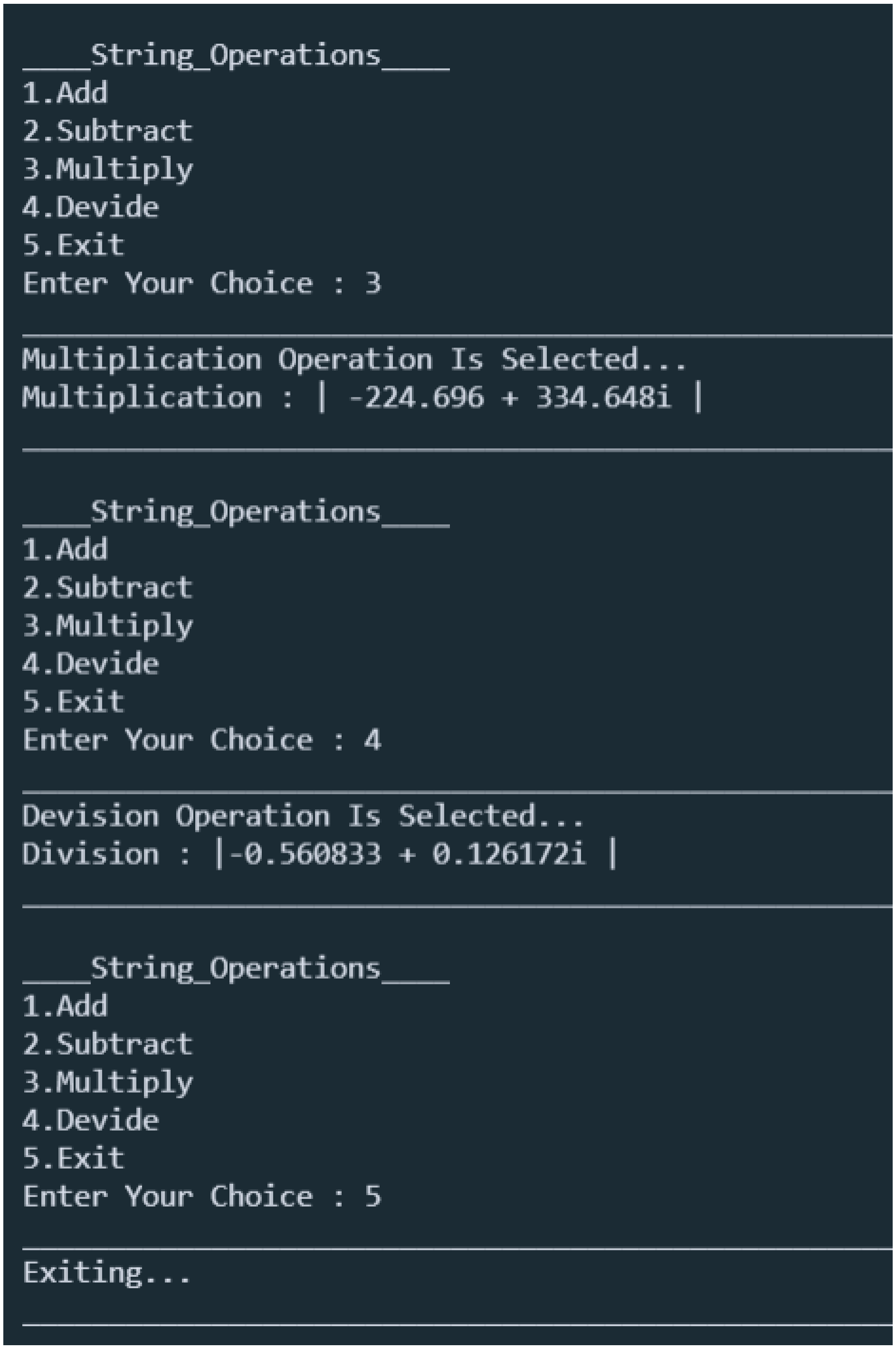
```
void Addition(float real[], float imaginary[])
{
    cout << "Addition Operation Is Selected..."
         << "\n";

    float r = real[0] + real[1], img = imaginary[0] +
imaginary[1];
    cout << "Addition : | " << r;
    if (img > 0)
        cout << " + ";
    cout << img << "i |\n";
}

void Subtract(float real[], float imaginary[])
{
    cout << "Subtraction Operation Is Selected..."
         << "\n";
    float r = real[0] - real[1], img = imaginary[0] -
imaginary[1];
    cout << "Subtraction : | " << r;
    if (img > 0)
        cout << " + ";
    cout << img << "i |\n";
}

void Multiply(float real[], float imaginary[])
{
    cout << "Multiplication Operation Is Selected..."
         << "\n";
    float r = real[0] * real[1] - imaginary[0] *
imaginary[1];
    float img = real[0] * imaginary[1] + imaginary[0] *
real[1];
    cout << "Multiplication : | " << r;
    if (img > 0)
        cout << " + ";
    cout << img << "i |\n";
}
```

```
void Division(float real[], float imaginary[])
{
    cout << "Devision Operation Is Selected..."
         << "\n";
    float devide = real[1] * real[1] + imaginary[1] *
imaginary[1];
    float r = real[0] * real[1] + imaginary[0] *
imaginary[1];
    float img = real[1] * imaginary[0] - imaginary[1] *
real[0];
    r /= devide;
    img /= devide;
    cout << "Division : |" << r;
    if (img > 0)
        cout << " + ";
    cout << img << "i |\n";
}

void Menu()
{
    cout << "\n____String_Operations____\n";
    cout << "1.Add\n";
    cout << "2.Subtract\n";
    cout << "3.Multiply\n";
    cout << "4.Devide\n";
    cout << "5.Exit\n";
    cout << "Enter Your Choice : ";
}

void AnsBar()
{
    cout <<
"____________________________________________________________
______\n";
}

bool Options(float real[], float imaginary[])
{
    int opt;
    fflush(stdin);
    cin >> opt;
    AnsBar();
    switch (opt)
    {
    case 1:
        Addition(real, imaginary);
        break;
    case 2:
        Subtract(real, imaginary);
        break;
    case 3:
        Multiply(real, imaginary);
        break;
    case 4:
        Division(real, imaginary);
        break;
    case 5:
        return 0;
    default:
        cout << "Invalid Input!\nTry Again!\n";
    }
    AnsBar();
    return 1;
}

int main()
{
    system("cls");
    cout << "_____Vicky_Gupta_20BCS070_____\n\n";
    char complex[100];
    cout << "Enter The Complex Number : \n";
    gets(complex);
    float real[2], imaginary[2];
    int itr = 0;
    int clen = lenght(complex);
    int complex_counter = 0;
    int prev_indx = 0;
    while (clen > itr)
    {
        int citr = itr; // complex iterator
        while (complex[citr] != ',' && complex[citr] !=
'\0')
            citr++;
        itr = citr + 1;
        citr--;
        char r[50], img[50];
        int iitr = 0, ritr = 0; // imaginary iterator real
iterator
        if (complex[citr] == 'i')
        {
            citr--;
            while (citr >= prev_indx && complex[citr] != '+'
&& complex[citr] != '-')
                img[iitr++] = complex[citr--];
            if (citr >= prev_indx)
            {
                img[iitr++] = complex[citr--];
            }
            img[iitr] = '\0';
            reverse(img);
        }
        else
            img[0] = '\0';
        if (citr > prev_indx)
        {
            while (citr >= prev_indx && complex[citr] != '+'
&& complex[citr] != '-')
                r[ritr++] = complex[citr--];
            if (citr >= prev_indx)
            {
                r[ritr++] = complex[citr--];
            }
            r[ritr] = '\0';
            reverse(r);
        }
        else
            r[0] = '\0';

        if (r[0] != '\0') // for real
        {
            bool isDecimal = check_decimal(r);
            if (isDecimal)
            {
                int decimal_index = 0;
                while (r[decimal_index] != '.')
                    decimal_index++;
                if (r[0] == '+' || r[0] == '-')
                    String_Integer(r, 1, decimal_index - 1,
real, complex_counter);
                else
                    String_Integer(r, 0, decimal_index - 1,
real, complex_counter);
                String_Decimal(r, decimal_index + 1,
lenght(r) - 1, real, complex_counter);
            }
            else
            {
                if (r[0] == '+' || r[0] == '-')
                    String_Integer(r, 1, lenght(r) - 1,
real, complex_counter);
                else
                    String_Integer(r, 0, lenght(r) - 1,
real, complex_counter);
            }
            if (r[0] == '-')
                real[complex_counter] = -
real[complex_counter];
        }
        else
            real[complex_counter] = 0;
        if (img[0] != '\0') // for imaginary
        {
            bool isDecimal = check_decimal(img);
            if (isDecimal)
            {
                int decimal_index = 0;
                while (img[decimal_index] != '.')
                    decimal_index++;
                if (img[0] == '+' || img[0] == '-')
                    String_Integer(img, 1, decimal_index - 
1, imaginary, complex_counter);
                else
                    String_Integer(img, 0, decimal_index - 
1, imaginary, complex_counter);
                String_Decimal(img, decimal_index + 1, 
lenght(img) - 1, imaginary, complex_counter);
            }
            else
            {
                if (img[0] == '+' || img[0] == '-')
                    String_Integer(img, 1, lenght(img) - 1, 
imaginary, complex_counter);
                else
                    String_Integer(img, 0, lenght(img) - 1, 
imaginary, complex_counter);
            }
            if (img[0] == '-')
                imaginary[complex_counter] = -
imaginary[complex_counter];
        }
        else
            imaginary[complex_counter] = 0;
        complex_counter++;
        prev_indx = itr;
    }
    cout << "\nComplex Number \n";
    for (int i = 0; i < 2; i++)
    {
        cout << i + 1 << ". " << real[i] << " " << 
imaginary[i] << "i\n";
    }
    cout << "\n";
    while (true)
    {
        Menu();
        if (!Options(real, imaginary))
            break;
    }
    cout << "Exiting...\n";
    AnsBar();
    return 0;
}
```

Output :-



```
_____Vicky_Gupta_20BCS070_____

Enter The Complex Number :
-12.56+8.6i,24.6-9.8i

Complex Number
1. -12.56 8.6i
2. 24.6 -9.8i


____String_Operations____
1.Add
2.Subtract
3.Multiply
4.Devide
5.Exit
Enter Your Choice : 1
_____________________________________________________
Addition Operation Is Selected...
Addition : | 12.04-1.2i |
_____________________________________________________

____String_Operations____
1.Add
2.Subtract
3.Multiply
4.Devide
5.Exit
Enter Your Choice : 2
_____________________________________________________
Subtraction Operation Is Selected...
Subtraction : | -37.16 + 18.4i |
_____________________________________________________
```



```
____String_Operations____
1.Add
2.Subtract
3.Multiply
4.Devide
5.Exit
Enter Your Choice : 3
___________________________________________________
Multiplication Operation Is Selected...
Multiplication : | -224.696 + 334.648i |
___________________________________________________

____String_Operations____
1.Add
2.Subtract
3.Multiply
4.Devide
5.Exit
Enter Your Choice : 4
___________________________________________________
Devision Operation Is Selected...
Division : |-0.560833 + 0.126172i |
___________________________________________________

____String_Operations____
1.Add
2.Subtract
3.Multiply
4.Devide
5.Exit
Enter Your Choice : 5
___________________________________________________
Exiting...
___________________________________________________
```