Department of Computer Science and Engineering (Data Science)

JHANVI PAREKH
60009210033
DATA SCIENCE
PLP

Aim: Implement Procedural Programming Using C Programming

Theory:

Procedural Programming;

Procedural Oriented Programming is a programming language that follows a step-by-step approach to break down a task into a collection of variables and routines (or subroutines) through a sequence of instructions. Each step is carried out in order in a systematic manner so that a computer can understand what to do.

C programing:

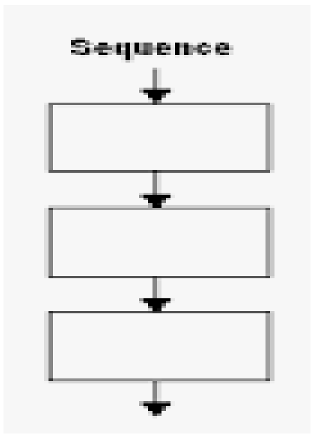
C is a powerful general-purpose programming language. It can be used to develop software like operating systems, databases, compilers, and so on. C programming is an excellent language to learn to program for beginners.

Control Structure

Control Structures are just a way to specify flow of control in programs. Any algorithm or program can be more clear and understood if they use self-contained modules called as logic or control structures. It basically analyses and chooses in which direction a program flows based on certain parameters or conditions.

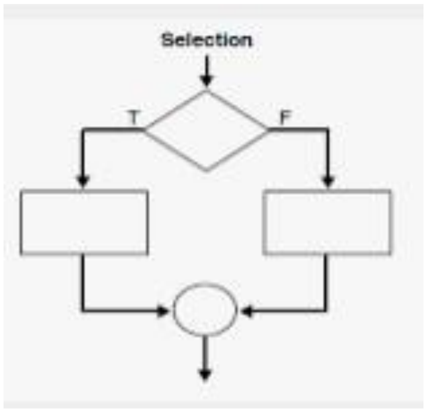
- **Sequential Control Structure**

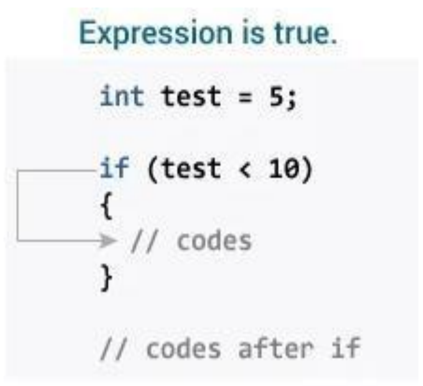
  Sequential logic as the name suggests follows a serial or sequential flow in which the flow depends on the series of instructions given to the computer.



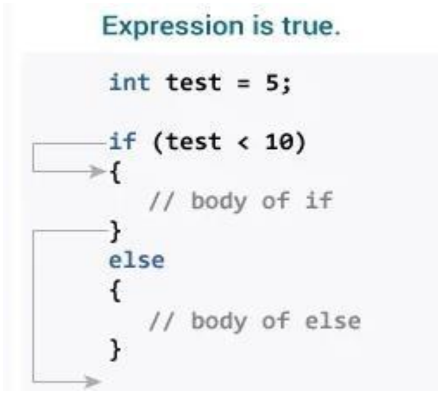
- **Conditional Control Structure**

  Selection Logic simply involves a number of conditions or parameters which decides one out of several written modules. The structures which use these type of logic are known as Conditional Structures.




Expression is true.

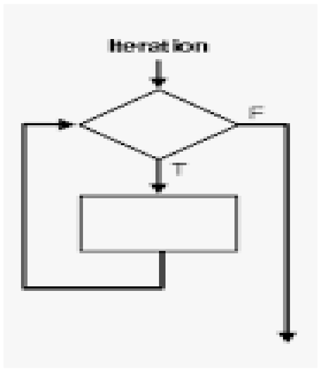
```
int test = 5;

if (test < 10)
{
   // codes
}

// codes after if
```


Expression is true.

```
int test = 5;

if (test < 10)
{
   // body of if
}
else
{
   // body of else
}
```

- **Iterative Control Structure**

  The Iteration logic employs a loop which involves a repeat statement followed by a module known as the body of a loop.



Example 1.1

Write a c Program to check whether Entered Year Is Leap or Not

```c
#include <stdio.h>
int main () {
  int year;
  printf ("Enter a year: ");
  scanf ("%d", &year);

  // leap year if perfectly divisible by 400
  if (year % 400 == 0) {
    printf ("%d is a leap year.", year);
  }
  // not a leap year if divisible by 100
  // but not divisible by 400
  else if (year % 100 == 0) {
    printf ("%d is not a leap year.", year);
  }
  // leap year if not divisible by 100
  // but divisible by 4
  else if (year % 4 == 0) {
    printf ("%d is a leap year.", year);
  }
  // all other years are not leap years
  else {
    printf ("%d is not a leap year.", year);
  }

  return 0;
}
```

```
Output:
Enter a year: 1900
1900 is not a leap year.

Enter a year: 2012
2012 is a leap year.
```

Array

An array is defined as the collection of similar type of data items stored at contiguous memory locations. Arrays are the derived data type in C programming language which can store the primitive type of data such as int, char, double, float, etc. It also has the capability to store the collection of derived data types, such as pointers, structure, etc. The array is the simplest data structure where each data element can be randomly accessed by using its index number.

Syntax Declaration of C Array

data_type array_name[array_size];

Example to declare the array.

int marks [5];

Here, int is the ***data_type*, marks** are the ***array_name***, and 5 is the ***array_size*.**

One Dimensional Array

One dimensional array is an array that has only one subscript specification that is needed to specify a particular element of an array. A one-dimensional array is a structured collection of components (often called array elements) that can be accessed individually by specifying the position of a component with a single index value.

Syntax: data-type arr_name[array_size];

Example: int n [5] = {0, 1, 2, 3, 4};

Two Dimensional Array

The two dimensional (2D) array in C programming is also known as matrix. A matrix can be represented as a table of rows and columns.

The syntax to declare the 2D array is given below.

data_type array_name[rows][columns];

Consider the following example.

int twodimen [4][3];

Example 1.2

Write a c Program to add all the elements of 1-dimension[1D] Array

```
#include<stdio.h>

int main () {
   int i, arr [50], sum, num;

   printf ("\nEnter no of elements:");
   scanf ("%d", &num);

   //Reading values into Array
   printf ("\nEnter the values:");
   for (i = 0; i < num; i++)
      scanf ("%d", &arr[i]);
```

```
Output:
Enter no of elements: 3
Enter the values: 11 22 33
a [0] =11
a [1] =22
a [2] =33
Sum=66
```

```
   //Computation of total
   sum = 0;
   for (i = 0; i < num; i++)
      sum = sum + arr[i];

   //Printing of all elements of array
   for (i = 0; i < num; i++)
      printf("\na[%d] =%d", i, arr[i]);

   //Printing of total
   printf ("\nSum=%d", sum);

   return (0);
 }
```

Pointer in C Programming

A pointer is a variable that stores the address of another variable. Unlike other variables that hold values of a certain type, pointer holds the address of a variable. For example, an integer variable holds (or you can say stores) an integer value, however an integer pointer holds the address of an integer variable.

Syntax for Pointer Declaration

datatype *pointer_name;

Example of The pointer

```
 int a = 10;
int *ptr; //pointer declaration
ptr = &a;//pointer initialization
```

Lab Assignments to complete in this session

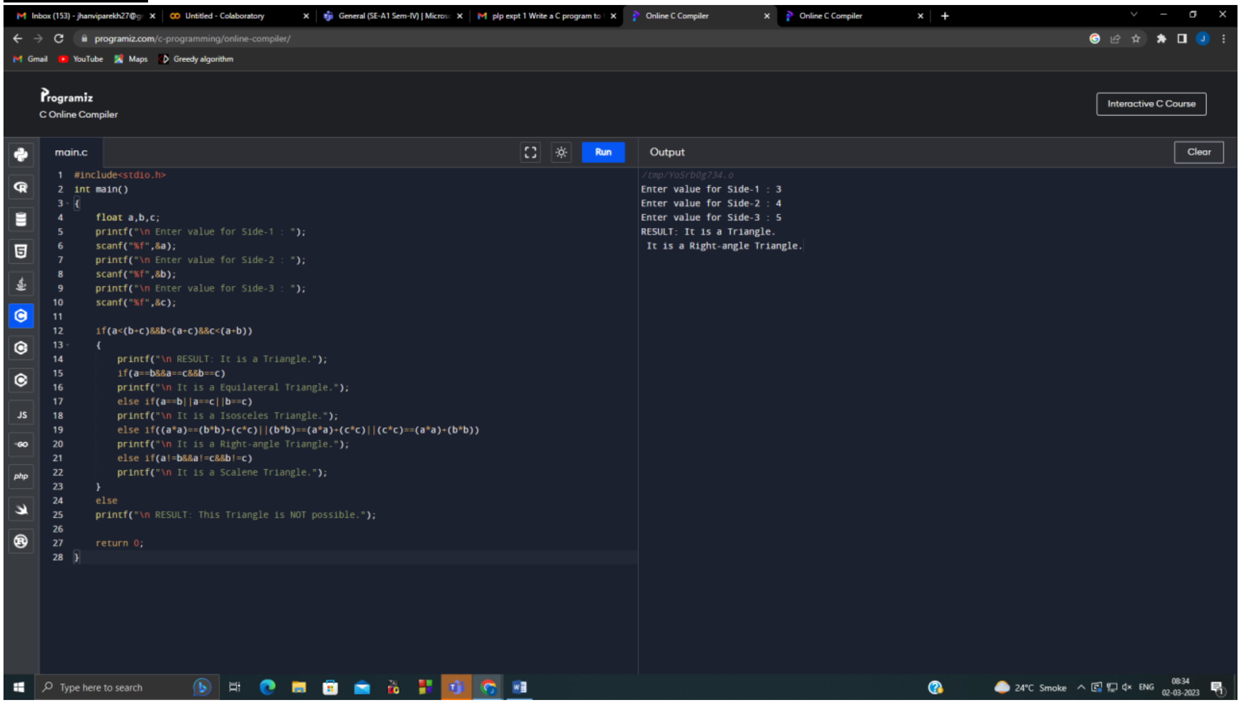
a) Write a C program to find whether a triangle can be formed or not. If not display “This Triangle is NOT possible.” If the triangle can be formed, then check whether the triangle formed is equilateral, isosceles, scalene or a right-angled triangle. (If it is a right-angled triangle then only print Right-angle triangle do not print it as Scalene Triangle or Isosceles triangle).

CODE:

```
#include<stdio.h>
int main()
{
   float a,b,c;
   printf("\n Enter value for Side-1 : ");
   scanf("%f",&a);
   printf("\n Enter value for Side-2 : ");
   scanf("%f",&b);
   printf("\n Enter value for Side-3 : ");
   scanf("%f",&c);

   if(a<(b+c)&&b<(a+c)&&c<(a+b))
```

```
    {
        printf("\n RESULT: It is a Triangle.");
        if(a==b&&a==c&&b==c)
        printf("\n It is a Equilateral Triangle.");
        else if(a==b||a==c||b==c)
        printf("\n It is a Isosceles Triangle.");
        else if((a*a)==(b*b)+(c*c)||(b*b)==(a*a)+(c*c)||(c*c)==(a*a)+(b*b))
        printf("\n It is a Right-angle Triangle.");
        else if(a!=b&&a!=c&&b!=c)
        printf("\n It is a Scalene Triangle.");
    }
    else
    printf("\n RESULT: This Triangle is NOT possible.");

    return 0;
}
```

OUTPUT:



```
#include<stdio.h>
int main()
{
    float a,b,c;
    printf("\n Enter value for Side-1 : ");
    scanf("%f",&a);
    printf("\n Enter value for Side-2 : ");
    scanf("%f",&b);
    printf("\n Enter value for Side-3 : ");
    scanf("%f",&c);

    if(a<(b+c)&&b<(a+c)&&c<(a+b))
    {
        printf("\n RESULT: It is a Triangle.");
        if(a==b&&a==c&&b==c)
        printf("\n It is a Equilateral Triangle.");
        else if(a==b||a==c||b==c)
        printf("\n It is a Isosceles Triangle.");
        else if((a*a)==(b*b)+(c*c)||(b*b)==(a*a)+(c*c)||(c*c)==(a*a)+(b*b))
        printf("\n It is a Right-angle Triangle.");
        else if(a!=b&&a!=c&&b!=c)
        printf("\n It is a Scalene Triangle.");
    }
    else
    printf("\n RESULT: This Triangle is NOT possible.");

    return 0;
}
```

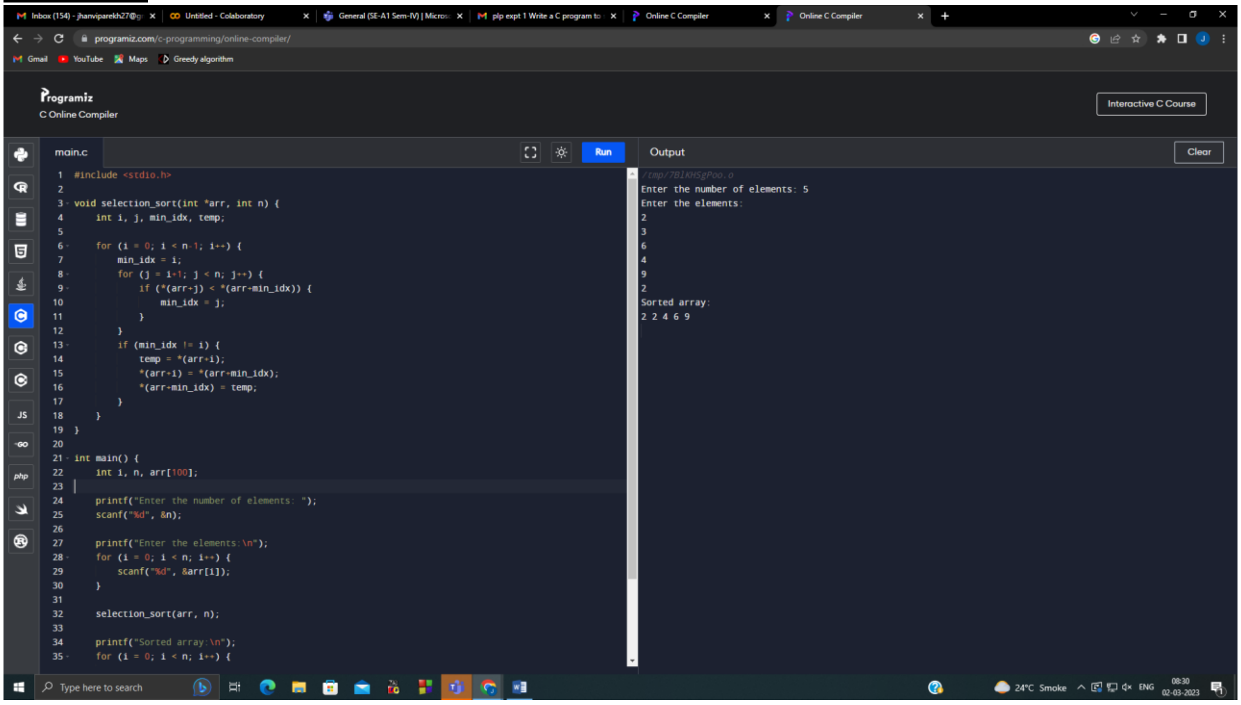
```
/tmp/YoSrb0g734.o
Enter value for Side-1 : 3
Enter value for Side-2 : 4
Enter value for Side-3 : 5
RESULT: It is a Triangle.
 It is a Right-angle Triangle.
```

b) Write a C program to sort a given 1D array using pointer in ascending order.

CODE:

```
#include <stdio.h>

void selection_sort(int *arr, int n) {
    int i, j, min_idx, temp;

    for (i = 0; i < n-1; i++) {
        min_idx = i;
        for (j = i+1; j < n; j++) {
            if (*(arr+j) < *(arr+min_idx)) {
                min_idx = j;
            }
        }
        if (min_idx != i) {
            temp = *(arr+i);
            *(arr+i) = *(arr+min_idx);
            *(arr+min_idx) = temp;
        }
    }
}

int main() {
    int i, n, arr[100];

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    selection_sort(arr, n);

    printf("Sorted array:\n");
    for (i = 0; i < n; i++) {
        printf("%d ", *(arr+i));
    }
    printf("\n");

    return 0;
}
```

OUTPUT:



```
#include <stdio.h>

void selection_sort(int *arr, int n) {
    int i, j, min_idx, temp;

    for (i = 0; i < n-1; i++) {
        min_idx = i;
        for (j = i+1; j < n; j++) {
            if (*(arr+j) < *(arr+min_idx)) {
                min_idx = j;
            }
        }
        if (min_idx != i) {
            temp = *(arr+i);
            *(arr+i) = *(arr+min_idx);
            *(arr+min_idx) = temp;
        }
    }
}

int main() {
    int i, n, arr[100];

    printf("Enter the number of elements: ");
    scanf("%d", &n);

    printf("Enter the elements:\n");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    selection_sort(arr, n);

    printf("Sorted array:\n");
    for (i = 0; i < n; i++) {
```

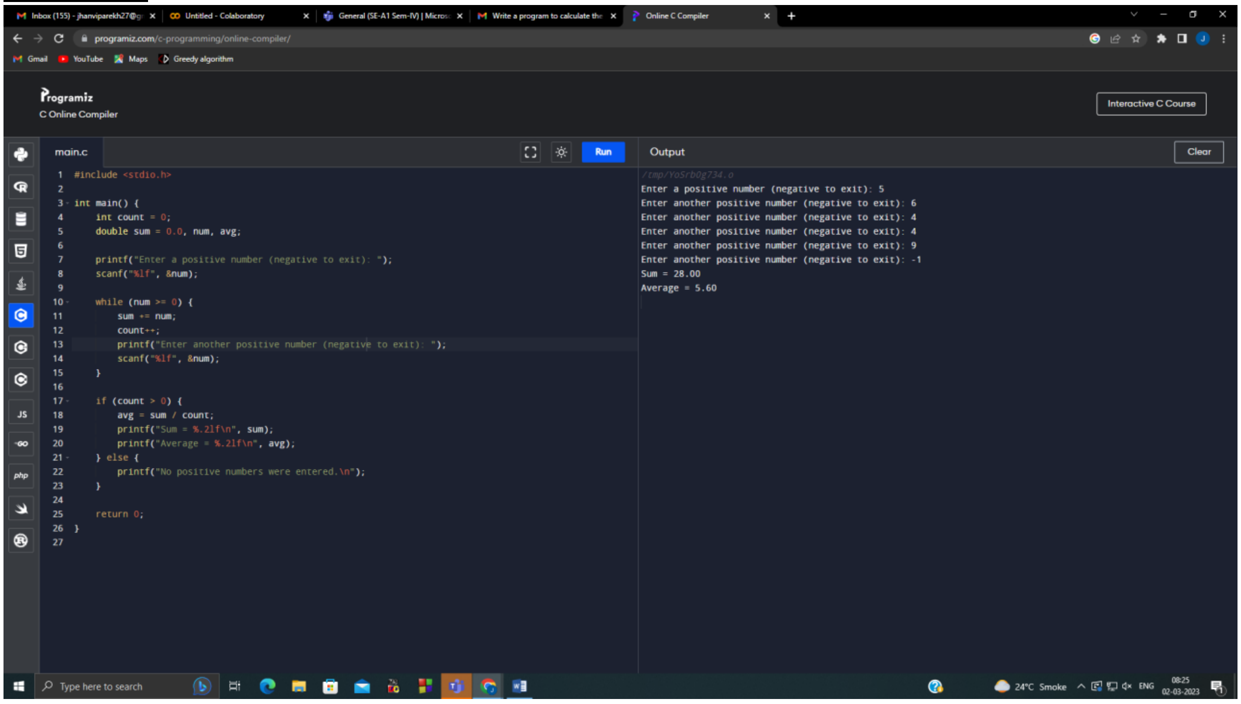
```
/tmp/7BlKHSgPoo.o
Enter the number of elements: 5
Enter the elements:
2
3
6
4
9
2
Sorted array:
2 2 4 6 9
```

c) Write a program to calculate the sum and average of positive numbers. If the user enters a negative number, the sum and average are displayed.(Go To)

CODE:

```
#include <stdio.h>

int main() {
   int count = 0;
   double sum = 0.0, num, avg;

   printf("Enter a positive number (negative to exit): ");
   scanf("%lf", &num);

   while (num >= 0) {
      sum += num;
      count++;
      printf("Enter another positive number (negative to exit): ");
      scanf("%lf", &num);
   }

   if (count > 0) {
      avg = sum / count;
      printf("Sum = %.2lf\n", sum);
      printf("Average = %.2lf\n", avg);
   } else {
      printf("No positive numbers were entered.\n");
   }

   return 0;
}
```

OUTPUT:



```
#include <stdio.h>

int main() {
    int count = 0;
    double sum = 0.0, num, avg;

    printf("Enter a positive number (negative to exit): ");
    scanf("%lf", &num);

    while (num >= 0) {
        sum += num;
        count++;
        printf("Enter another positive number (negative to exit): ");
        scanf("%lf", &num);
    }

    if (count > 0) {
        avg = sum / count;
        printf("Sum = %.2lf\n", sum);
        printf("Average = %.2lf\n", avg);
    } else {
        printf("No positive numbers were entered.\n");
    }

    return 0;
}
```

```
/tmp/YoSrb0g734.o
Enter a positive number (negative to exit): 5
Enter another positive number (negative to exit): 6
Enter another positive number (negative to exit): 4
Enter another positive number (negative to exit): 4
Enter another positive number (negative to exit): 9
Enter another positive number (negative to exit): -1
Sum = 28.00
Average = 5.60
```

d) Write a Menu driven Program to create a simple calculator (perform addition Subtraction multiplication division).

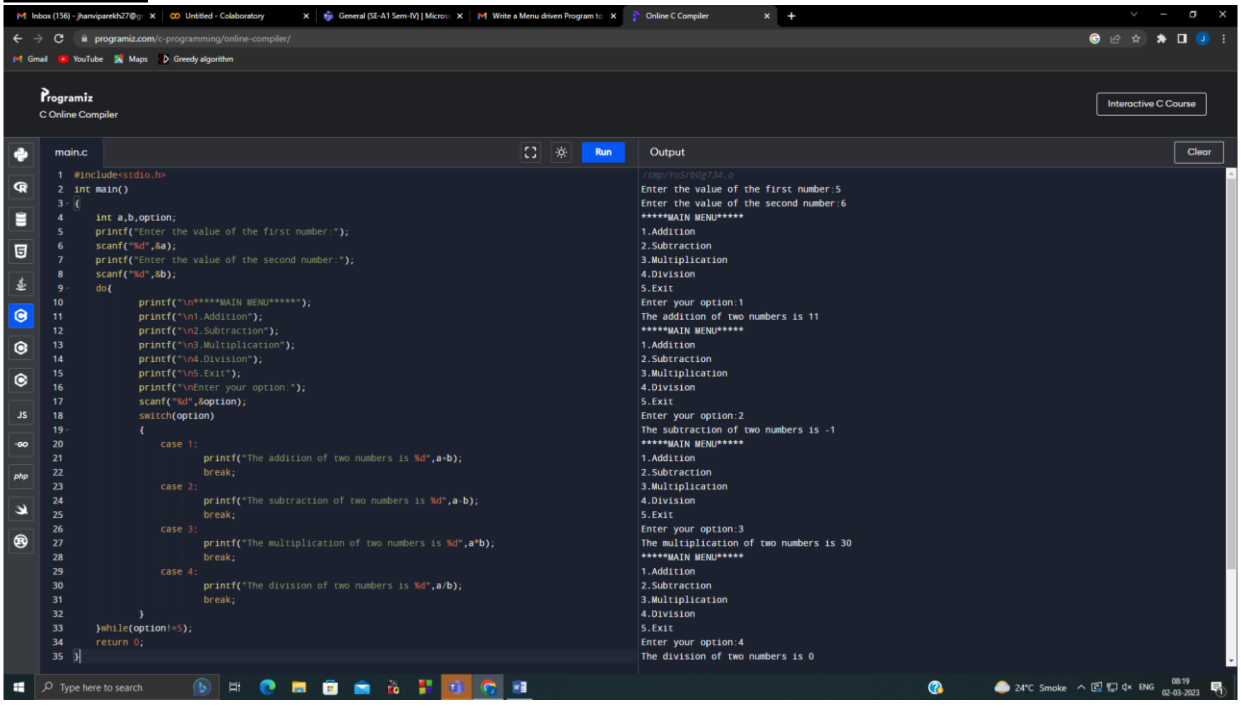
CODE:

```
#include<stdio.h>
int main()
{
   int a,b,option;
   printf("Enter the value of the first number:");
   scanf("%d",&a);
   printf("Enter the value of the second number:");
   scanf("%d",&b);
   do{
         printf("\n*****MAIN MENU*****");
         printf("\n1.Addition");
         printf("\n2.Subtraction");
         printf("\n3.Multiplication");
         printf("\n4.Division");
         printf("\n5.Exit");
         printf("\nEnter your option:");
         scanf("%d",&option);
         switch(option)
         {
            case 1:
                  printf("The addition of two numbers is %d",a+b);
                  break;
            case 2:
                  printf("The subtraction of two numbers is %d",a-b);
                  break;
            case 3:
                  printf("The multiplication of two numbers is %d",a*b);
                  break;
            case 4:
                  printf("The division of two numbers is %d",a/b);
                  break;
         }
   }while(option!=5);
   return 0;
}
```

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

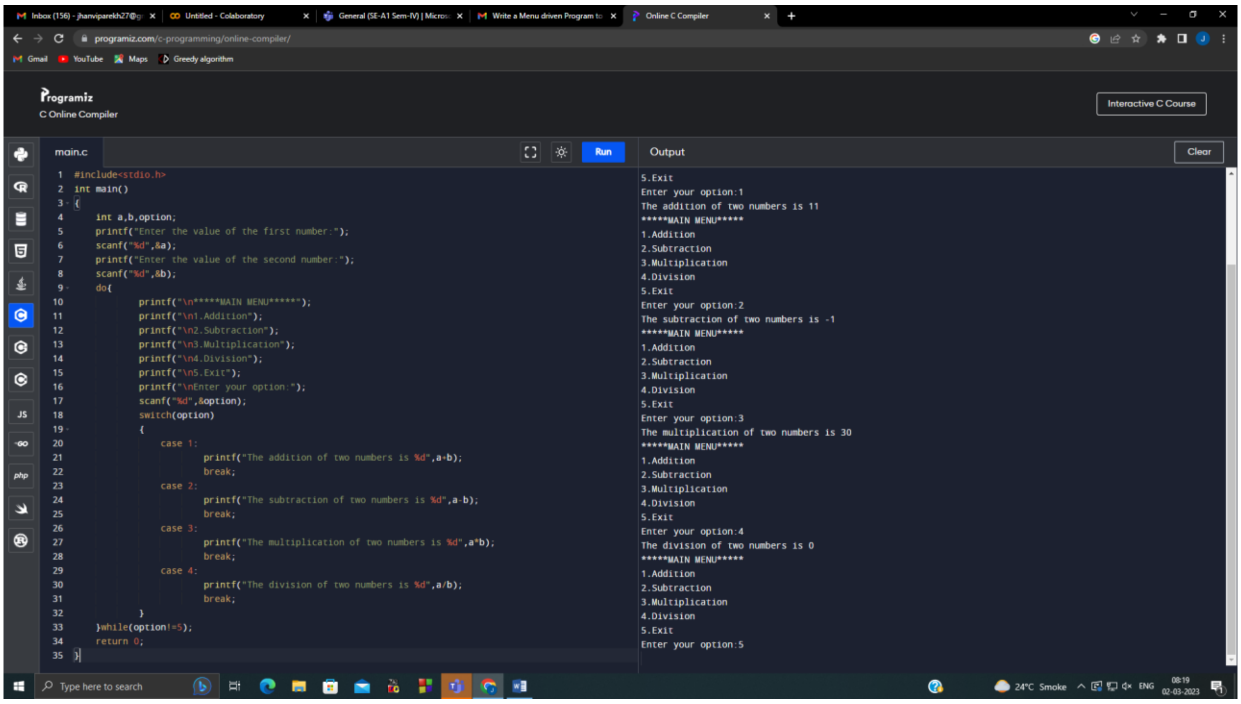
Department of Computer Science and Engineering (Data Science)

OUTPUT:



```c
#include<stdio.h>
int main()
{
    int a,b,option;
    printf("Enter the value of the first number:");
    scanf("%d",&a);
    printf("Enter the value of the second number:");
    scanf("%d",&b);
    do{
            printf("\n*****MAIN MENU*****");
            printf("\n1.Addition");
            printf("\n2.Subtraction");
            printf("\n3.Multiplication");
            printf("\n4.Division");
            printf("\n5.Exit");
            printf("\nEnter your option:");
            scanf("%d",&option);
            switch(option)
            {
                case 1:
                        printf("The addition of two numbers is %d",a+b);
                        break;
                case 2:
                        printf("The subtraction of two numbers is %d",a-b);
                        break;
                case 3:
                        printf("The multiplication of two numbers is %d",a*b);
                        break;
                case 4:
                        printf("The division of two numbers is %d",a/b);
                        break;
            }
    }while(option!=5);
    return 0;
}
```

```
/tmp/YoSrb0g734.o
Enter the value of the first number:5
Enter the value of the second number:6
*****MAIN MENU*****
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your option:1
The addition of two numbers is 11
*****MAIN MENU*****
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your option:2
The subtraction of two numbers is -1
*****MAIN MENU*****
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your option:3
The multiplication of two numbers is 30
*****MAIN MENU*****
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your option:4
The division of two numbers is 0
```



```c
#include<stdio.h>
int main()
{
    int a,b,option;
    printf("Enter the value of the first number:");
    scanf("%d",&a);
    printf("Enter the value of the second number:");
    scanf("%d",&b);
    do{
            printf("\n*****MAIN MENU*****");
            printf("\n1.Addition");
            printf("\n2.Subtraction");
            printf("\n3.Multiplication");
            printf("\n4.Division");
            printf("\n5.Exit");
            printf("\nEnter your option:");
            scanf("%d",&option);
            switch(option)
            {
                case 1:
                        printf("The addition of two numbers is %d",a+b);
                        break;
                case 2:
                        printf("The subtraction of two numbers is %d",a-b);
                        break;
                case 3:
                        printf("The multiplication of two numbers is %d",a*b);
                        break;
                case 4:
                        printf("The division of two numbers is %d",a/b);
                        break;
            }
    }while(option!=5);
    return 0;
}
```

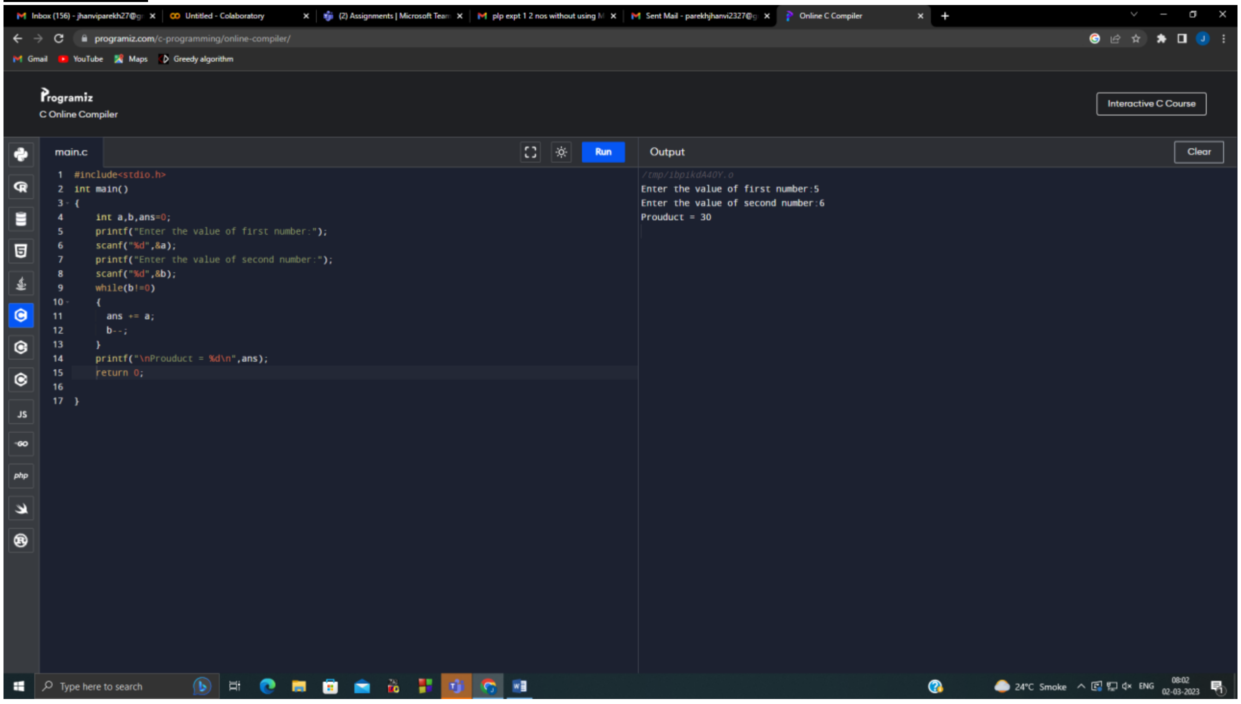
```
5.Exit
Enter your option:1
The addition of two numbers is 11
*****MAIN MENU*****
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your option:2
The subtraction of two numbers is -1
*****MAIN MENU*****
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your option:3
The multiplication of two numbers is 30
*****MAIN MENU*****
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your option:4
The division of two numbers is 0
*****MAIN MENU*****
1.Addition
2.Subtraction
3.Multiplication
4.Division
5.Exit
Enter your option:5
```

e) Write a c Program to perform Multiplication of 2 nos without using Multiplication Operator

CODE:

```c
#include<stdio.h>
int main()
{
    int a,b,ans=0;
    printf("Enter the value of first number:");
    scanf("%d",&a);
    printf("Enter the value of second number:");
    scanf("%d",&b);
    while(b!=0)
    {
      ans += a;
      b--;
    }
    printf("\nProuduct = %d\n",ans);
    return 0;
}
```

OUTPUT:



```
#include<stdio.h>
int main()
{
    int a,b,ans=0;
    printf("Enter the value of first number:");
    scanf("%d",&a);
    printf("Enter the value of second number:");
    scanf("%d",&b);
    while(b!=0)
    {
      ans += a;
      b--;
    }
    printf("\nProuduct = %d\n",ans);
    return 0;

}
```

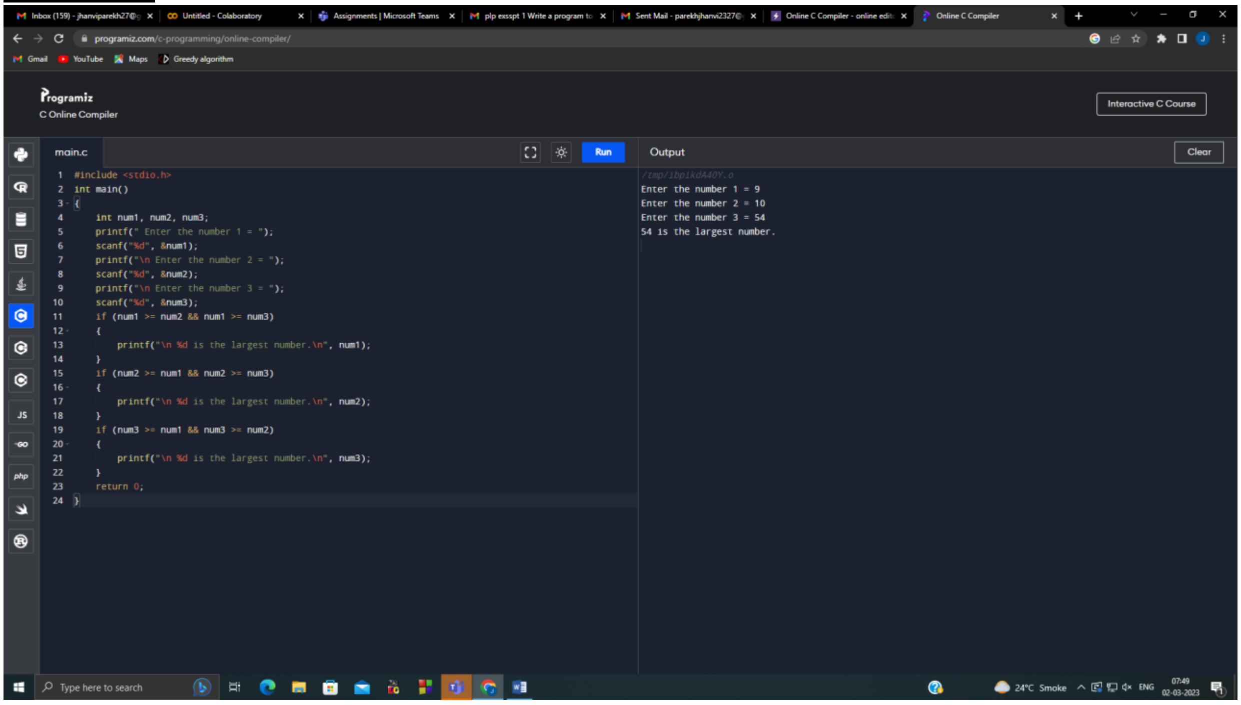
```
/tmp/ibpikdA40Y.o
Enter the value of first number:5
Enter the value of second number:6
Prouduct = 30
```

f) Write a program to find maximum of 3 numbers

CODE:

```
#include <stdio.h>
int main()
{
    int num1, num2, num3;
    printf(" Enter the number 1 = ");
    scanf("%d", &num1);
    printf("\n Enter the number 2 = ");
    scanf("%d", &num2);
    printf("\n Enter the number 3 = ");
    scanf("%d", &num3);
    if (num1 >= num2 && num1 >= num3)
    {
        printf("\n %d is the largest number.\n", num1);
    }
    if (num2 >= num1 && num2 >= num3)
    {
        printf("\n %d is the largest number.\n", num2);
    }
    if (num3 >= num1 && num3 >= num2)
    {
        printf("\n %d is the largest number.\n", num3);
    }
    return 0;
}
```

OUTPUT:



```
#include <stdio.h>
int main()
{
    int num1, num2, num3;
    printf(" Enter the number 1 = ");
    scanf("%d", &num1);
    printf("\n Enter the number 2 = ");
    scanf("%d", &num2);
    printf("\n Enter the number 3 = ");
    scanf("%d", &num3);
    if (num1 >= num2 && num1 >= num3)
    {
        printf("\n %d is the largest number.\n", num1);
    }
    if (num2 >= num1 && num2 >= num3)
    {
        printf("\n %d is the largest number.\n", num2);
    }
    if (num3 >= num1 && num3 >= num2)
    {
        printf("\n %d is the largest number.\n", num3);
    }
    return 0;
}
```

```
/tmp/ibpikdA40Y.o
Enter the number 1 = 9
Enter the number 2 = 10
Enter the number 3 = 54
54 is the largest number.
```

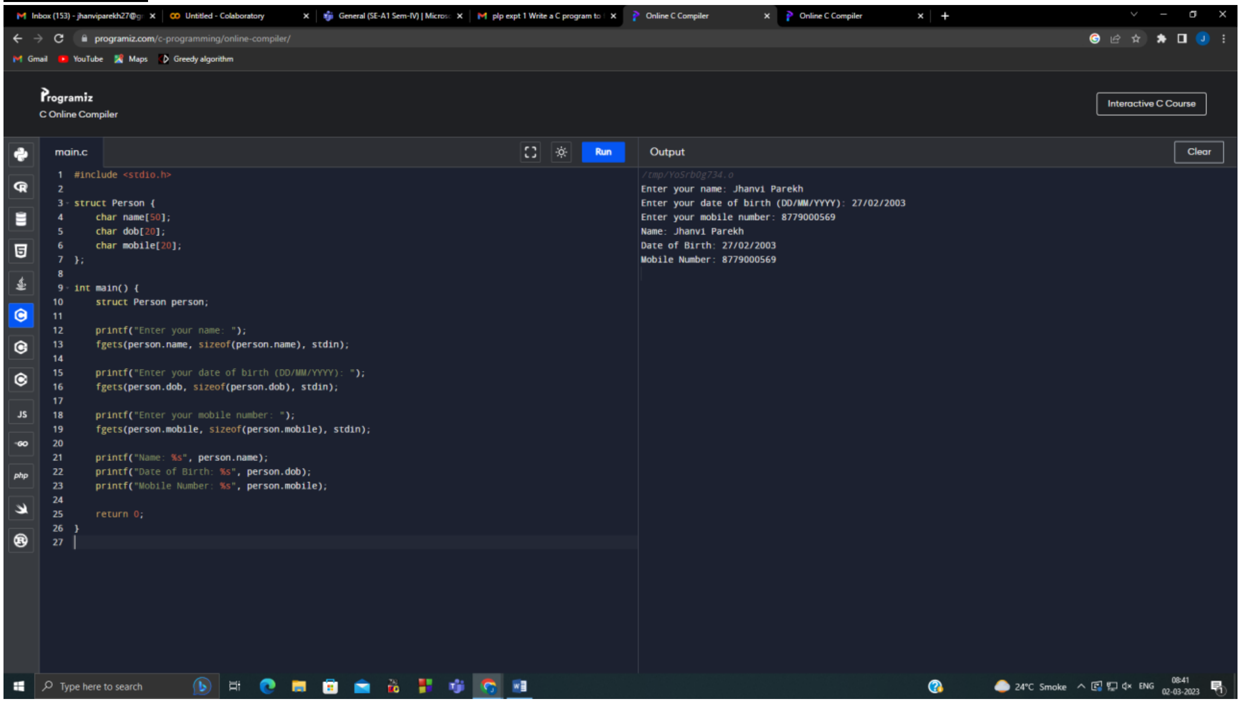
g) Write a C program to print your name, date of birth. and mobile number
Expected Output: (Use Structure)

```
Name: Alexandra Abramov
DOB: July 14, 1975
Mobile: 99-9999999999
```

CODE:

```
#include <stdio.h>

struct Person {
    char name[50];
    char dob[20];
    char mobile[20];
};

int main() {
    struct Person person;

    printf("Enter your name: ");
    fgets(person.name, sizeof(person.name), stdin);
    printf("Enter your date of birth (DD/MM/YYYY): ");
    fgets(person.dob, sizeof(person.dob), stdin);

    printf("Enter your mobile number: ");
    fgets(person.mobile, sizeof(person.mobile), stdin);

    printf("Name: %s", person.name);
    printf("Date of Birth: %s", person.dob);
    printf("Mobile Number: %s", person.mobile);

    return 0;
}
```

OUTPUT:



```
#include <stdio.h>

struct Person {
    char name[50];
    char dob[20];
    char mobile[20];
};

int main() {
    struct Person person;

    printf("Enter your name: ");
    fgets(person.name, sizeof(person.name), stdin);

    printf("Enter your date of birth (DD/MM/YYYY): ");
    fgets(person.dob, sizeof(person.dob), stdin);

    printf("Enter your mobile number: ");
    fgets(person.mobile, sizeof(person.mobile), stdin);

    printf("Name: %s", person.name);
    printf("Date of Birth: %s", person.dob);
    printf("Mobile Number: %s", person.mobile);

    return 0;
}
```

```
/tmp/YoSrb0g734.o
Enter your name: Jhanvi Parekh
Enter your date of birth (DD/MM/YYYY): 27/02/2003
Enter your mobile number: 8779000569
Name: Jhanvi Parekh
Date of Birth: 27/02/2003
Mobile Number: 8779000569
```

h) Write a C program to convert prefix expression to postfix expression.

Examples:
Input: Prefix: *+AB-CD
Output: Postfix: AB+CD-*
Explanation: Prefix to Infix: (A+B) * (C-D)
Infix to Postfix: AB+CD-*

Input : Prefix : *-A/BC-/AKL
Output : Postfix : ABC/-AK/L-*
Explanation : Prefix to Infix : (A-(B/C))*((A/K)-L)
Infix to Postfix : ABC/-AK/L-*

CODE:

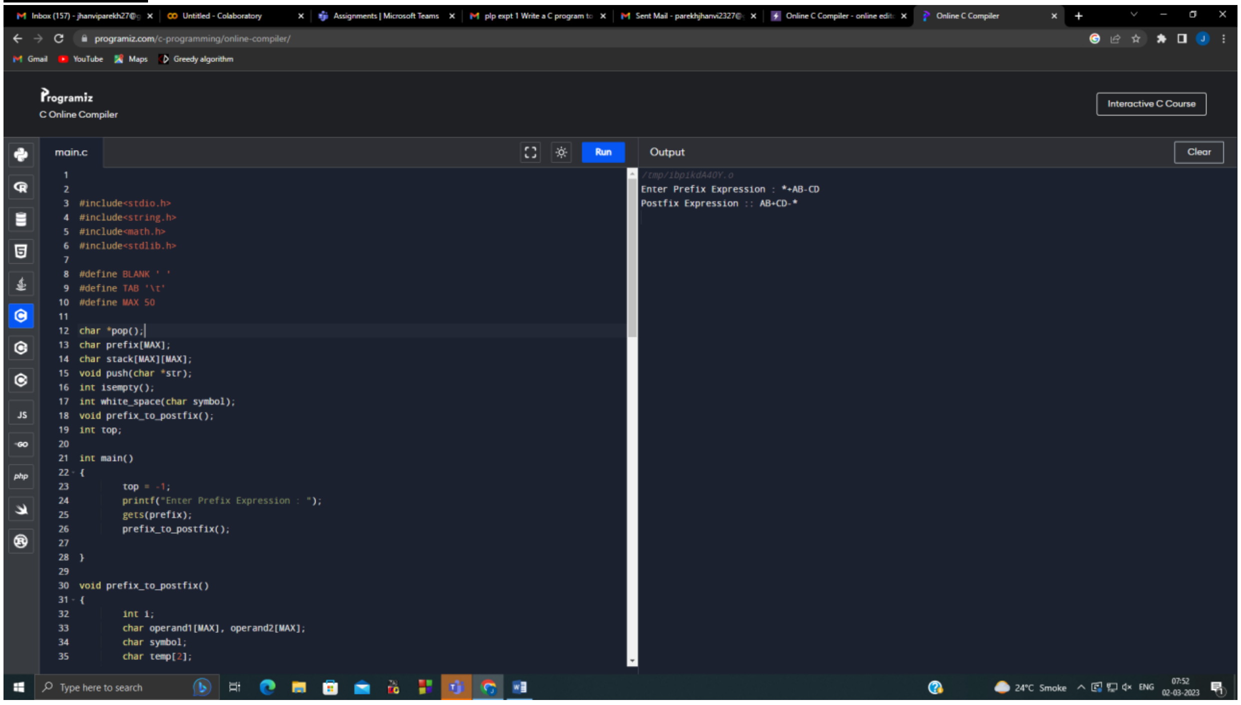
```
#include<stdio.h>
#include<string.h>
```

```
#include<math.h>
#include<stdlib.h>

#define BLANK ' '
#define TAB '\t'
#define MAX 50

char *pop();
char prefix[MAX];
char stack[MAX][MAX];
void push(char *str);
int isempty();
int white_space(char symbol);
void prefix_to_postfix();
int top;

int main()
{
        top = -1;
        printf("Enter Prefix Expression : ");
        gets(prefix);
        prefix_to_postfix();

}

void prefix_to_postfix()
{
        int i;
        char operand1[MAX], operand2[MAX];
        char symbol;
        char temp[2];
        char strin[MAX];
        for(i=strlen(prefix)-1;i>=0;i--)
        {
                symbol=prefix[i];
                temp[0]=symbol;
                temp[1]='\0';

                if(!white_space(symbol))
                {
                        switch(symbol)
                        {
                        case '+':
                        case '-':
                        case '*':
                        case '/':
                        case '%':
                        case '^':
```

```
                    strcpy(operand1,pop());
                    strcpy(operand2,pop());
                    strcpy(strin,operand1);
                    strcat(strin,operand2);
                    strcat(strin,temp);
                    push(strin);
                    break;
                default:
                    push(temp);
                }
            }
    }
    printf("\nPostfix Expression :: ");
    puts(stack[0]);
}

void push(char *str)
{
    if(top > MAX)
    {
        printf("\nStack overflow\n");
        exit(1);
    }
    else
    {
        top=top+1;
        strcpy( stack[top], str);
    }
}

char *pop()
{
    if(top == -1 )
    {
        printf("\nStack underflow \n");
        exit(2);
    }
    else
        return (stack[top--]);
}
int isempty()
{
    if(top==-1)
        return 1;
    else
        return 0;
}
int white_space(char symbol)
```

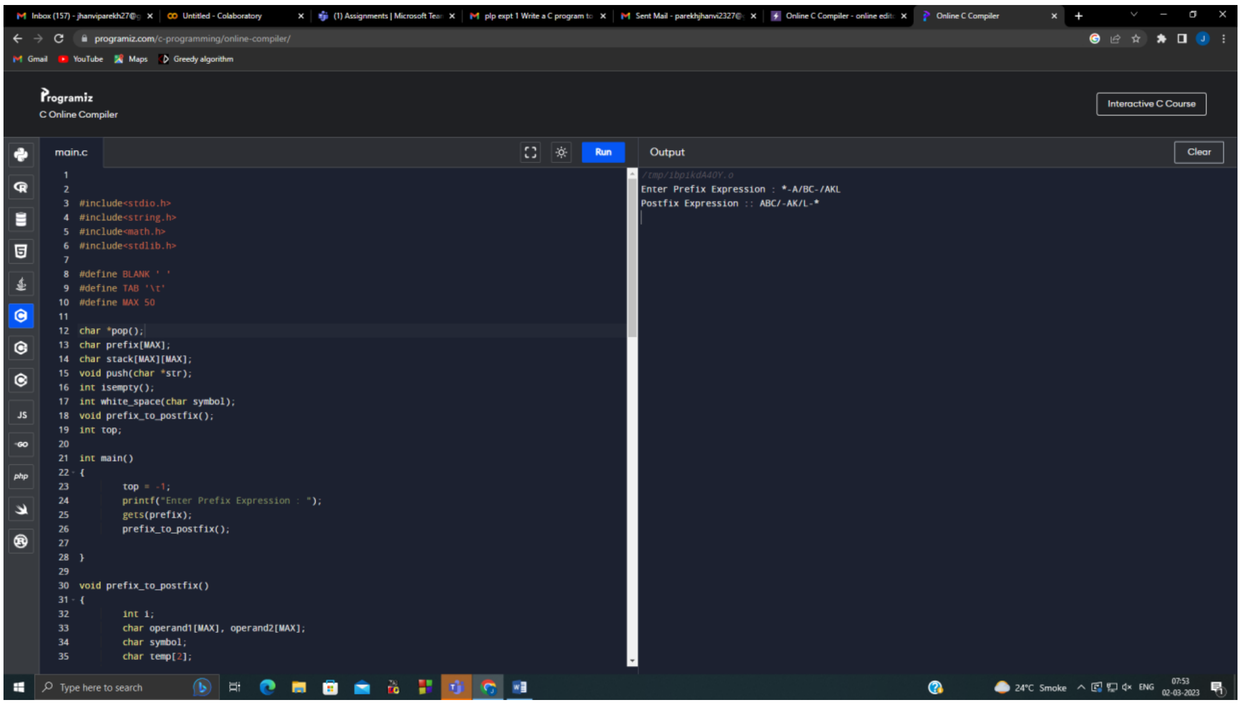
```
{
        if(symbol==BLANK || symbol==TAB || symbol=='\0')
                return 1;
        else
                return 0;
}
```

OUTPUT:



```
#include<stdio.h>
#include<string.h>
#include<math.h>
#include<stdlib.h>

#define BLANK ' '
#define TAB '\t'
#define MAX 50

char *pop();
char prefix[MAX];
char stack[MAX][MAX];
void push(char *str);
int isempty();
int white_space(char symbol);
void prefix_to_postfix();
int top;

int main()
{
        top = -1;
        printf("Enter Prefix Expression : ");
        gets(prefix);
        prefix_to_postfix();

}

void prefix_to_postfix()
{
        int i;
        char operand1[MAX], operand2[MAX];
        char symbol;
        char temp[2];
```

```
/tmp/ibpikdA40Y.o
Enter Prefix Expression : *+AB-CD
Postfix Expression :: AB+CD-*
```



```
/tmp/ibpikdA40Y.o
Enter Prefix Expression : *-A/BC-/AKL
Postfix Expression :: ABC/-AK/L-*
```