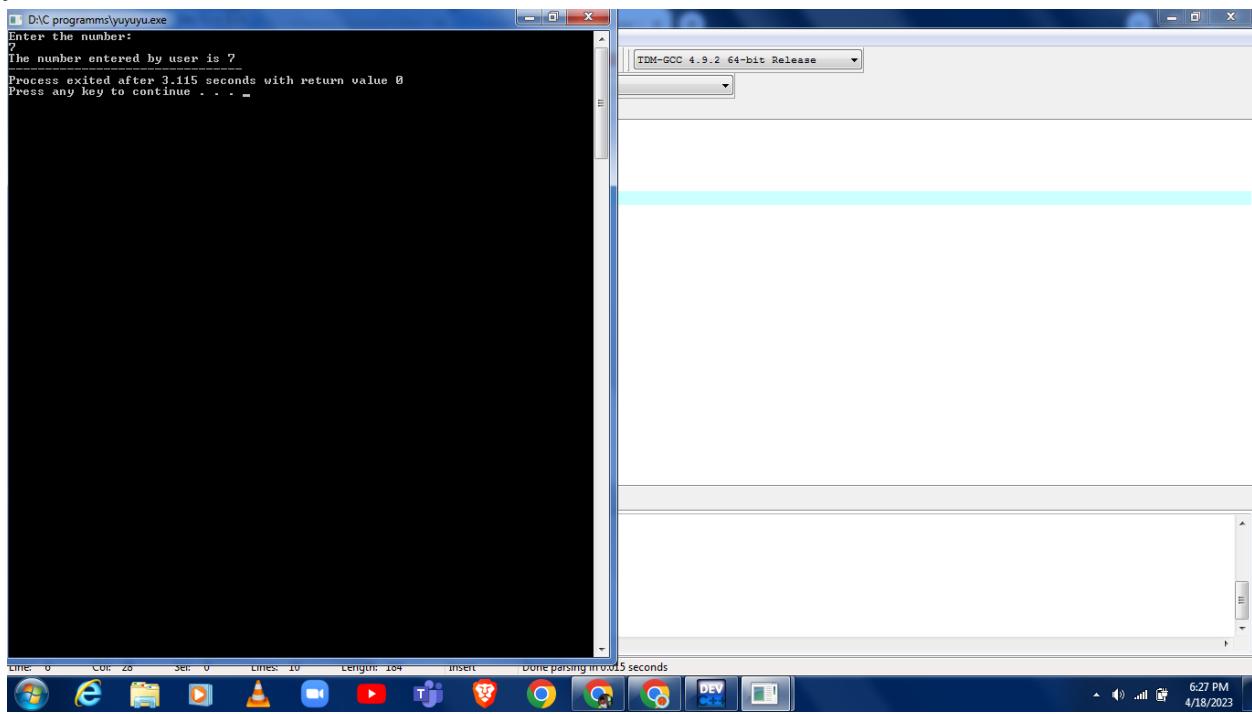


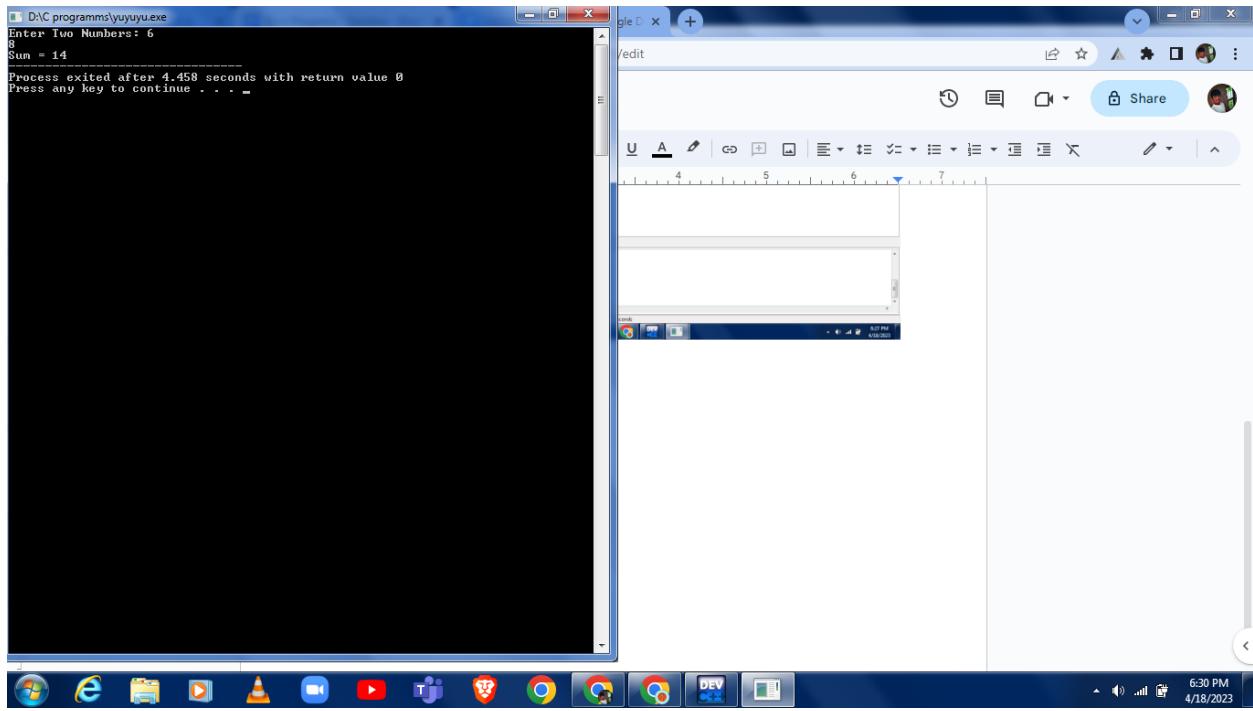
1.entered by user

```
#include <iostream>
using namespace std;
int main()
{
    int num;
    cout<<"Enter the number:";
    cin>>num;
    cout<<"The number entered by user is "<<num;
    return 0;
}
```



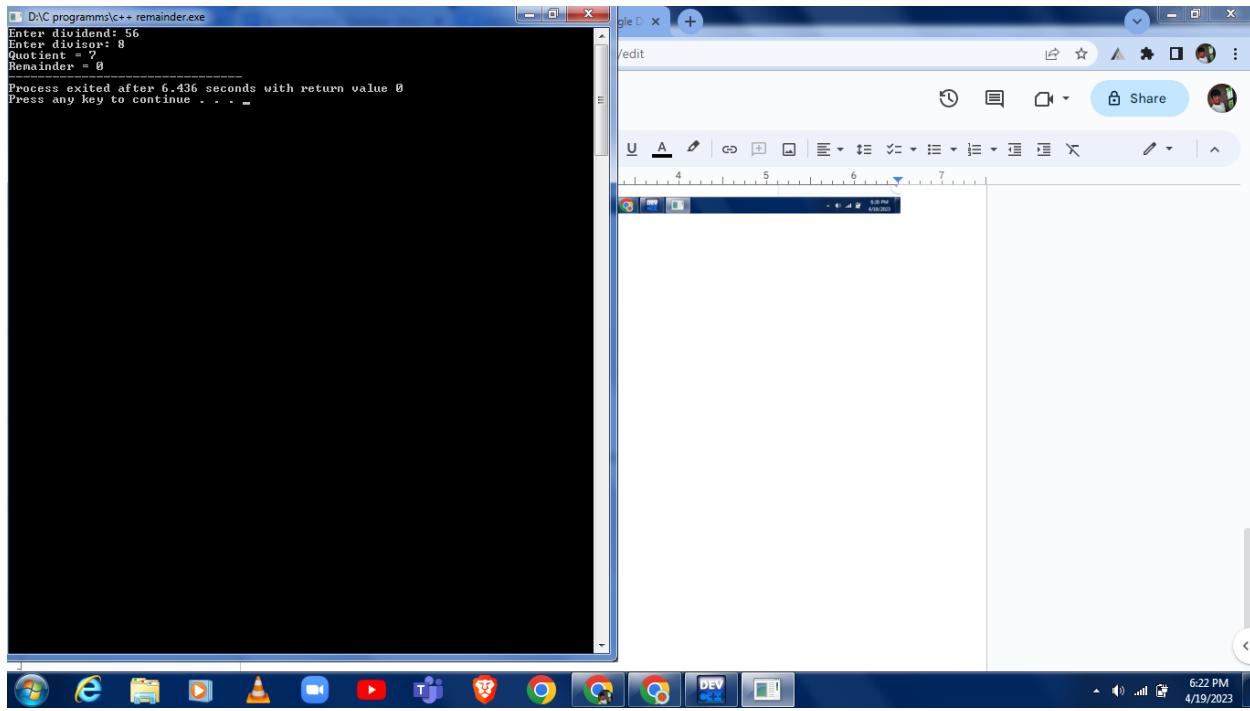
2.addition of two numbers

```
#include<iostream>
using namespace std;
int main()
{
    int num1,num2,add;
    cout<<"Enter Two Numbers: ";
    cin>>num1>>num2;
    add=num1+num2;
    cout<<"Sum = "<<add;
    return 0;
}
```



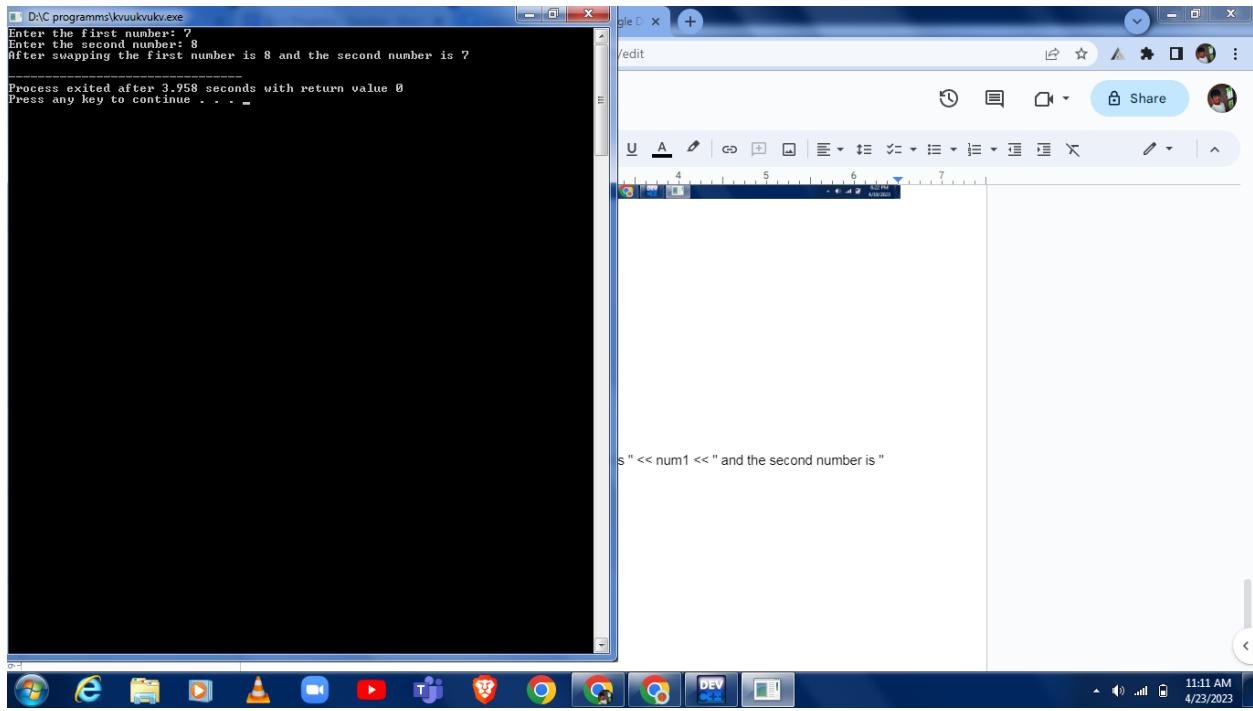
3. Find quotient and remainder

```
#include <iostream>
using namespace std;
int main()
{
    int divisor, dividend, quotient, remainder;
    cout << "Enter dividend: ";
    cin >> dividend;
    cout << "Enter divisor: ";
    cin >> divisor;
    quotient = dividend / divisor;
    remainder = dividend % divisor;
    cout << "Quotient = " << quotient << endl;
    cout << "Remainder = " << remainder;
    return 0;
}
```



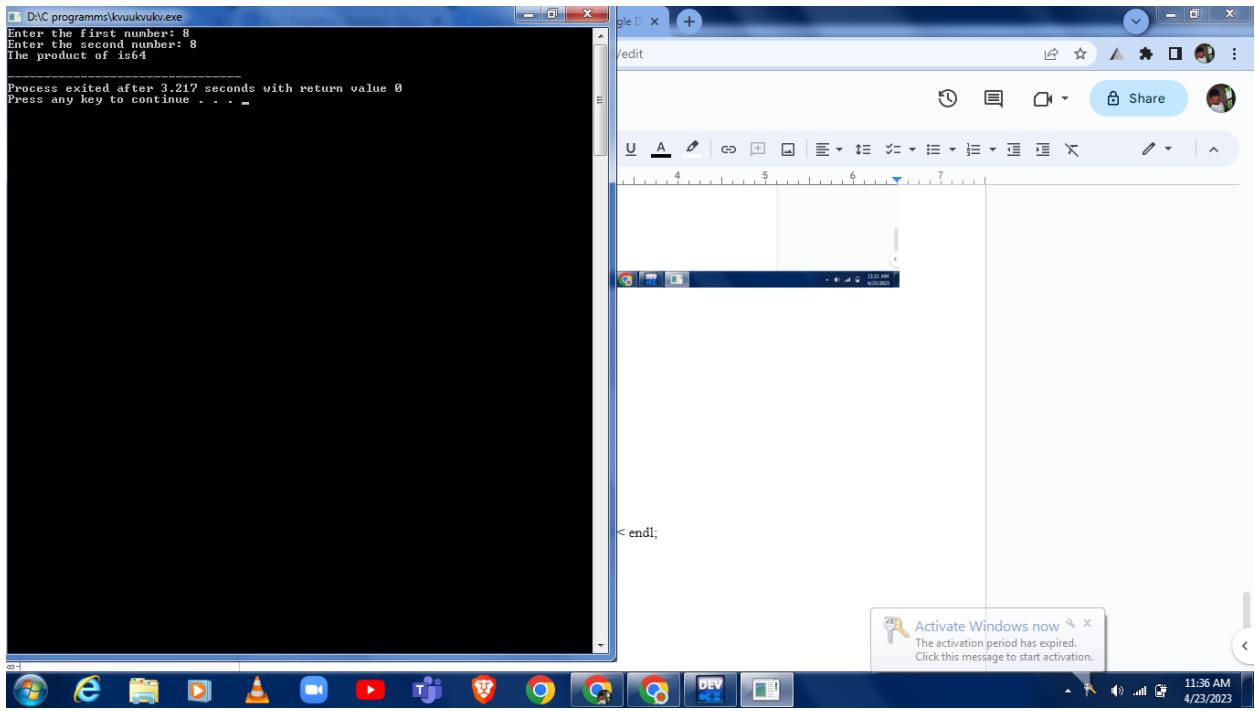
4. swap two num without using 3rd variable

```
#include <iostream>
using namespace std;
int main()
{
    int num1, num2;
    cout << "Enter the first number: ";
    cin >> num1;
    cout << "Enter the second number: ";
    cin >> num2;
    num1 = num1 + num2;
    num2 = num1 - num2;
    num1 = num1 - num2;
    cout << "After swapping the first number is " << num1 << " and the second number is "
<< num2 << endl;
    return 0;
}
```



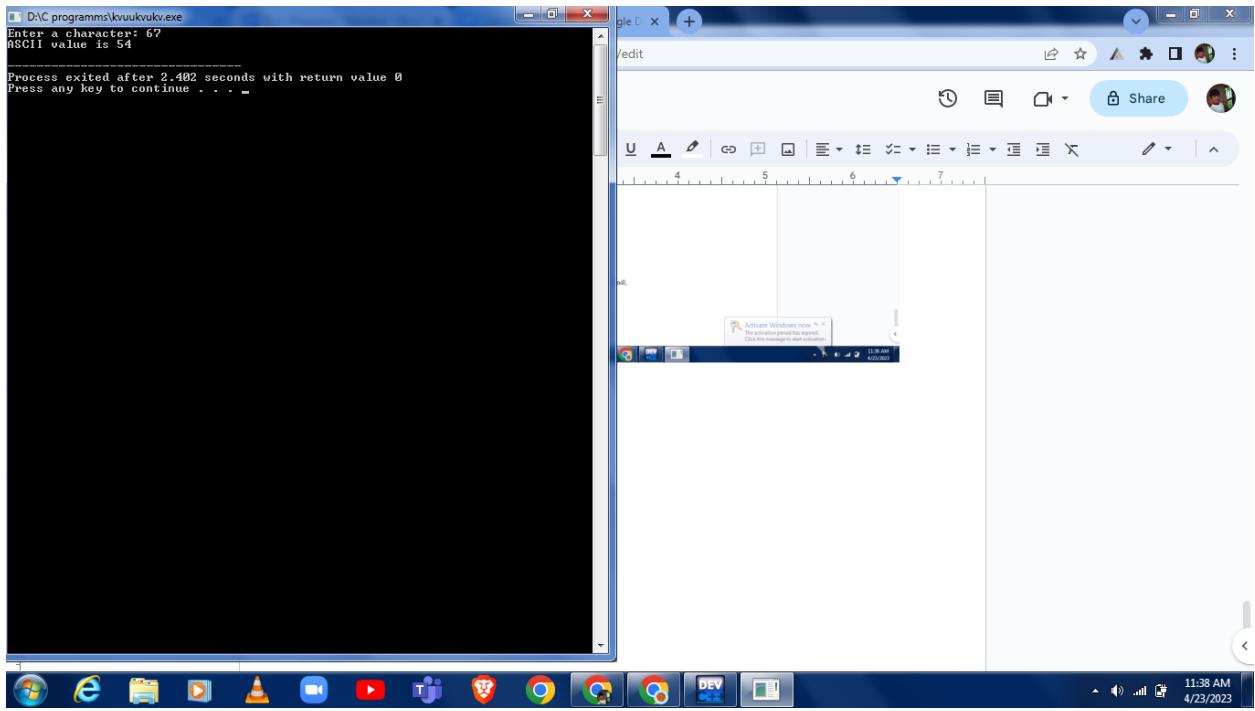
5.product of two numbers

```
#include <iostream>
using namespace std;
int main()
{
    float num1, num2, product;
    cout << "Enter the first number: ";
    cin >> num1;
    cout << "Enter the second number: ";
    cin >> num2;
    product = num1 * num2;
    cout << "The product of is " << product << endl;
    return 0;
}
```



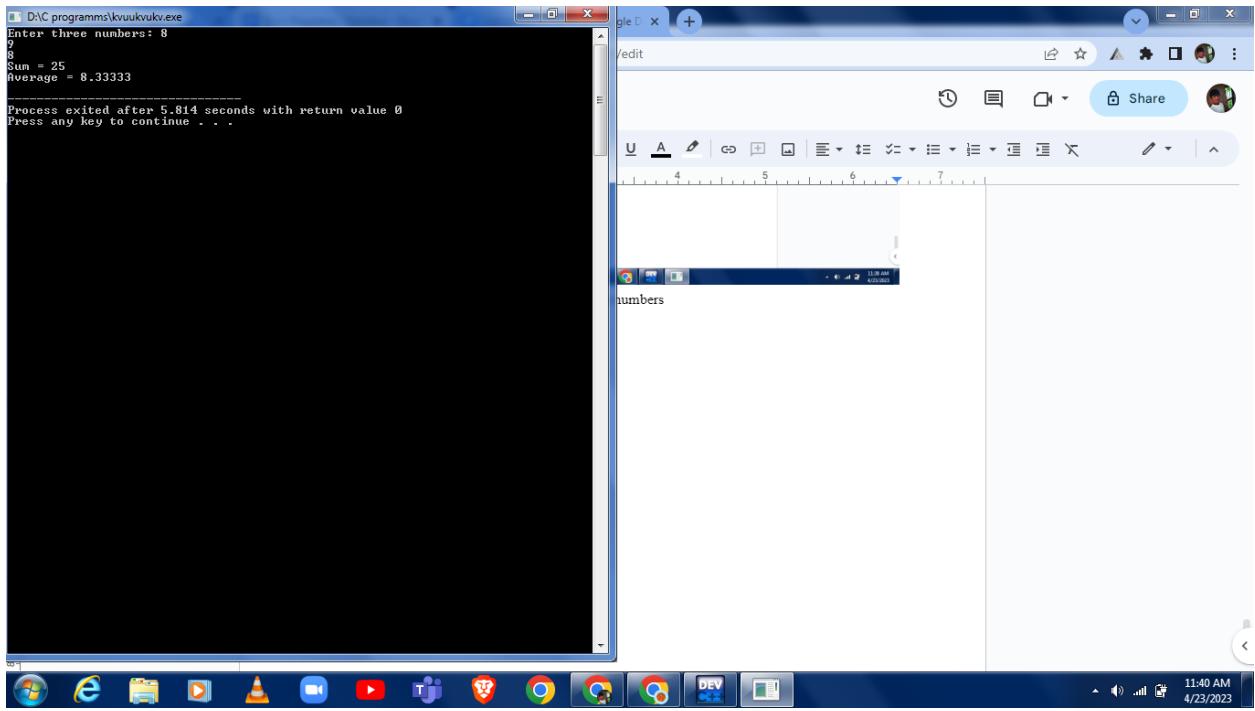
6. Program to Find ASCII Value of a Character

```
#include <iostream>
using namespace std;
int main()
{
    char c;
    cout << "Enter a character: ";
    cin >> c;
    cout << "ASCII value is " << int(c) << endl;
    return 0;
}
```



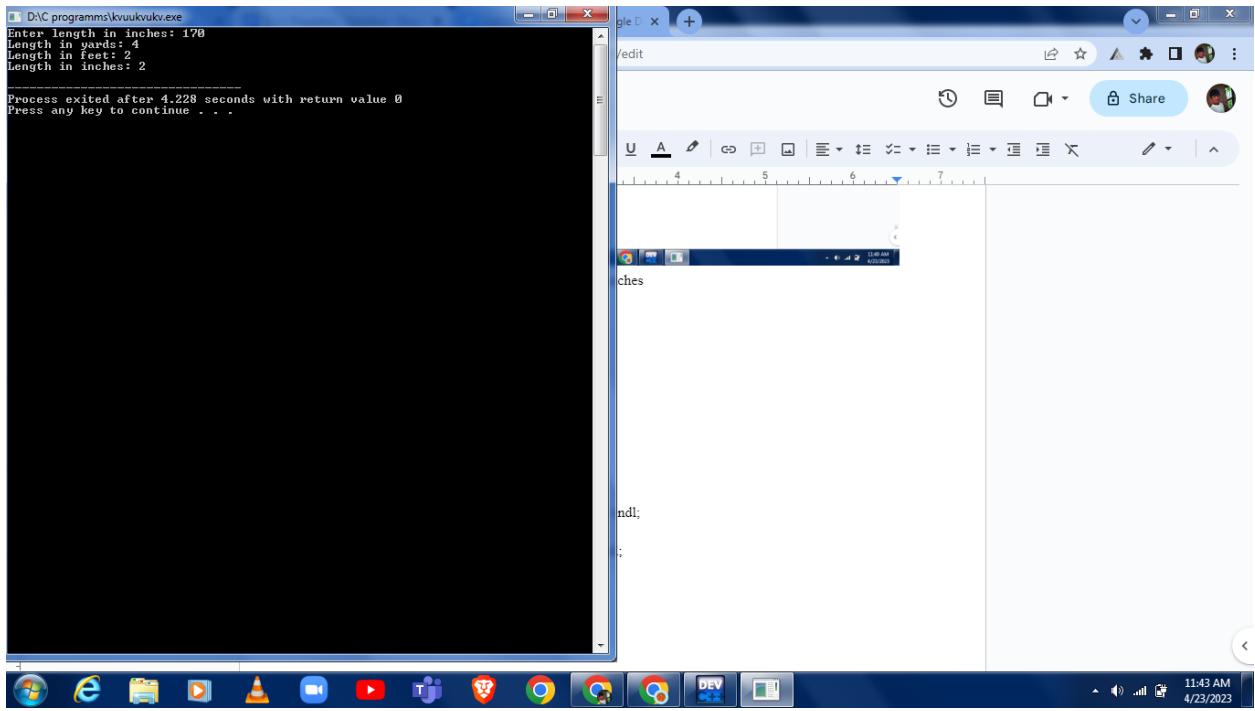
7. Program to calculate sum and average of three numbers

```
#include <iostream>
using namespace std;
int main()
{
    int num1,num2,num3;
    float sum,average;
    cout << "Enter three numbers: ";
    cin >> num1 >> num2 >> num3;
    sum = num1 + num2 + num3;
    average = sum / 3;
    cout << "Sum = " << sum << endl;
    cout << "Average = " << average << endl;
    return 0;
}
```



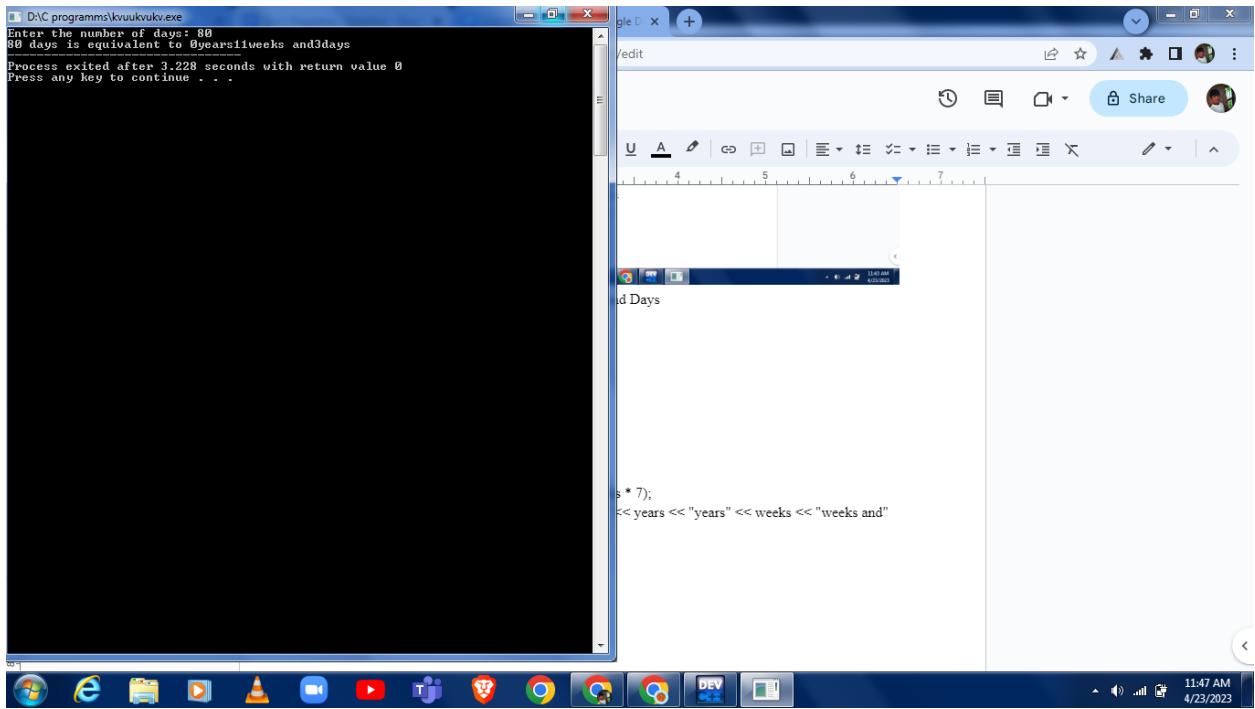
8. Program to convert inches into yard, feet and inches

```
#include <iostream>
using namespace std;
int main()
{
    int inches;
    cout << "Enter length in inches: ";
    cin >> inches;
    int yards = inches / 36;
    inches %= 36;
    int feet = inches / 12;
    inches %= 12;
    cout << "Length in yards: " << yards << endl;
    cout << "Length in feet: " << feet << endl;
    cout << "Length in inches: " << inches << endl;
    return 0;
}
```



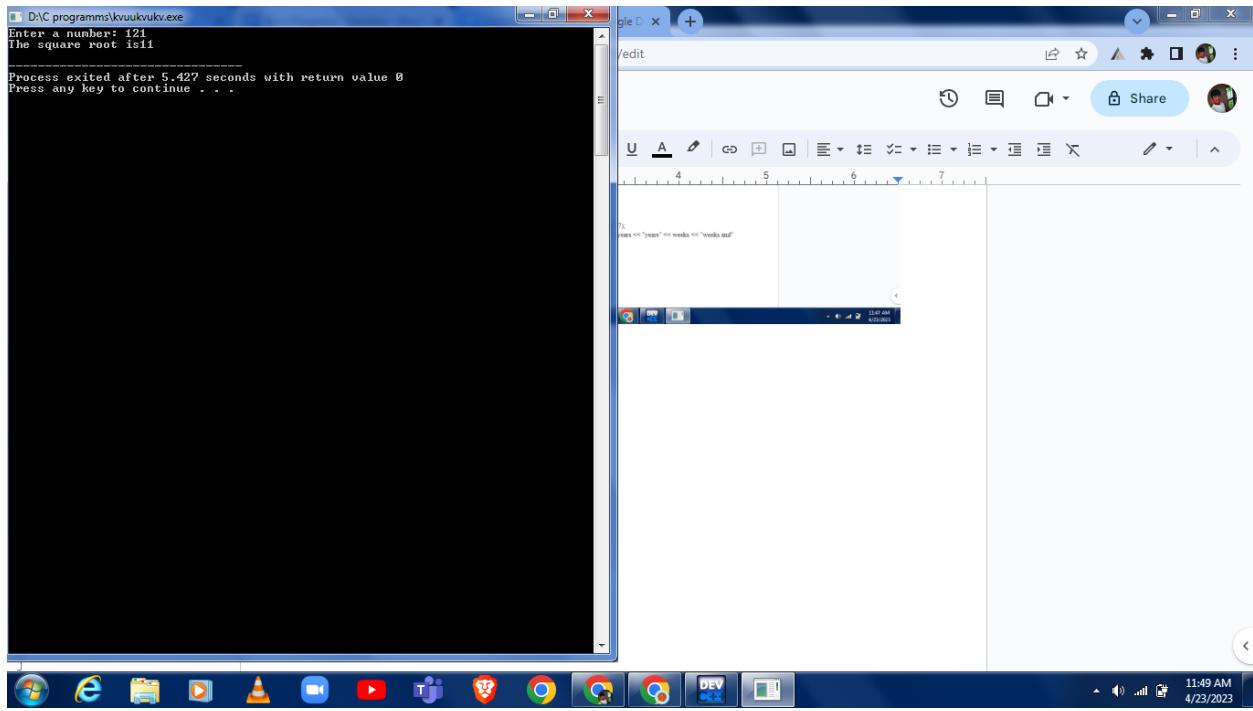
9. Program to Convert Days Into Years, Weeks and Days

```
#include <iostream>
using namespace std;
int main()
{
    int days,years,weeks,remaining_days;
    cout << "Enter the number of days: ";
    cin >> days;
    years = days / 365;
    weeks = (days % 365) / 7;
    remaining_days = days - (years * 365) - (weeks * 7);
    cout << days << " days is equivalent to " << years << "years" << weeks << "weeks and"
    << remaining_days << "days";
    return 0;
}
```



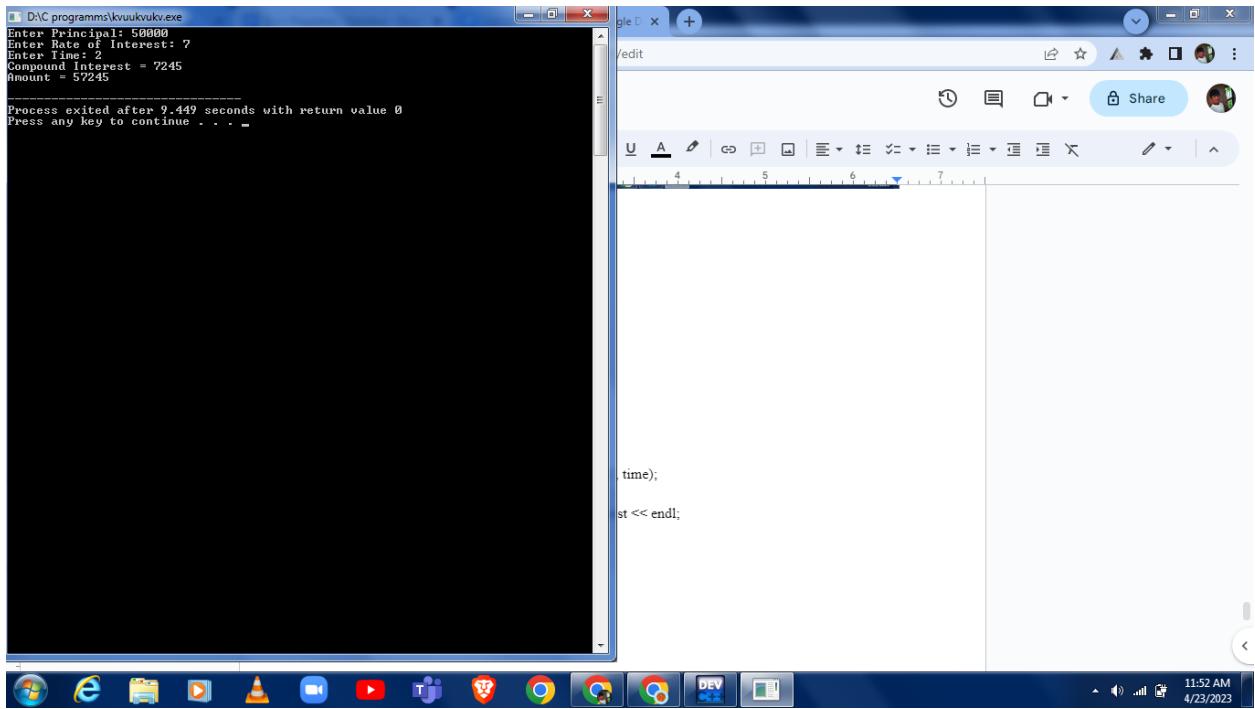
10. square root of a number

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    double num,ans;
    cout << "Enter a number: ";
    cin >> num;
    ans = sqrt(num);
    cout << "The square root is" << ans << endl;
    return 0;
}
```



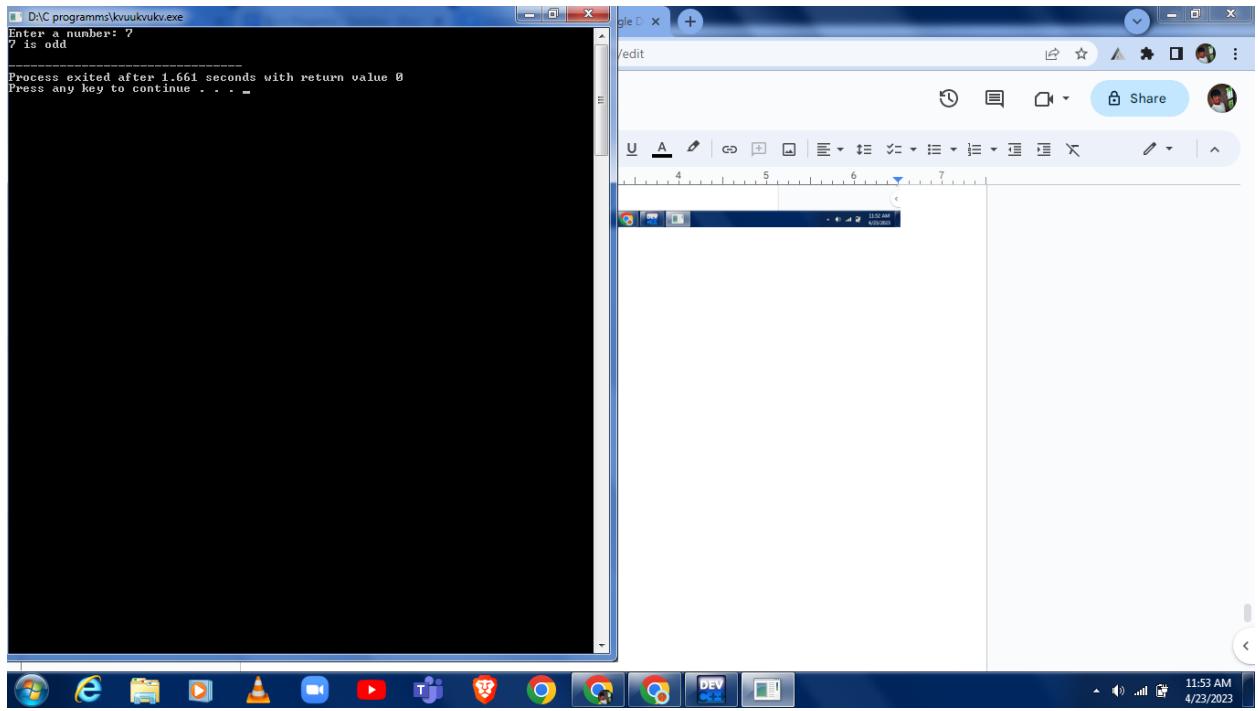
11. Program to find Compound Interest

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    float principal,rate,time,interest,amount;
    cout << "Enter Principal: ";
    cin >> principal;
    cout << "Enter Rate of Interest: ";
    cin >> rate;
    cout << "Enter Time: ";
    cin >> time;
    amount = principal * pow(1 + (rate / 100), time);
    interest = amount - principal;
    cout << "Compound Interest = " << interest << endl;
    cout << "Amount = " << amount << endl;
    return 0;
}
```



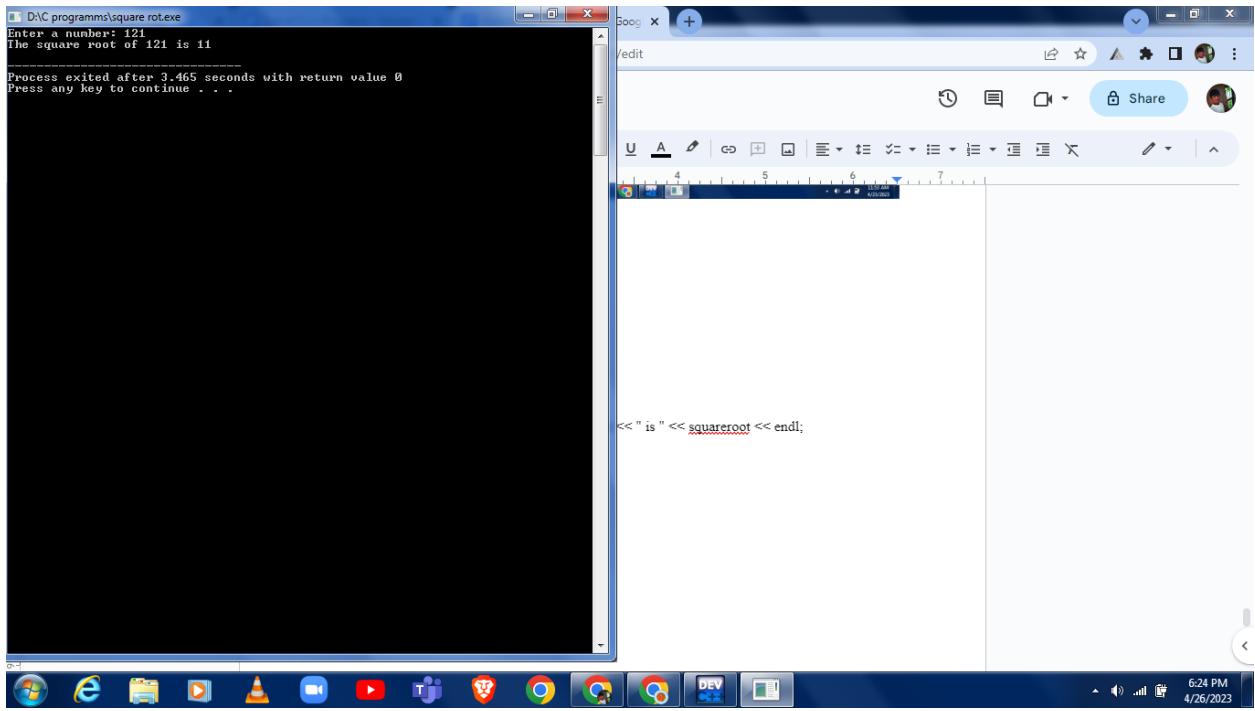
12. Program to Check Number is Odd or Even

```
#include <iostream>
using namespace std;
int main()
{
    int number;
    cout << "Enter a number: ";
    cin >> number;
    if(number % 2 == 0)
    {
        cout << number << " is even" << endl;
    } else
    {
        cout << number << " is odd" << endl;
    }
    return 0;
}
```



13.square root of a num

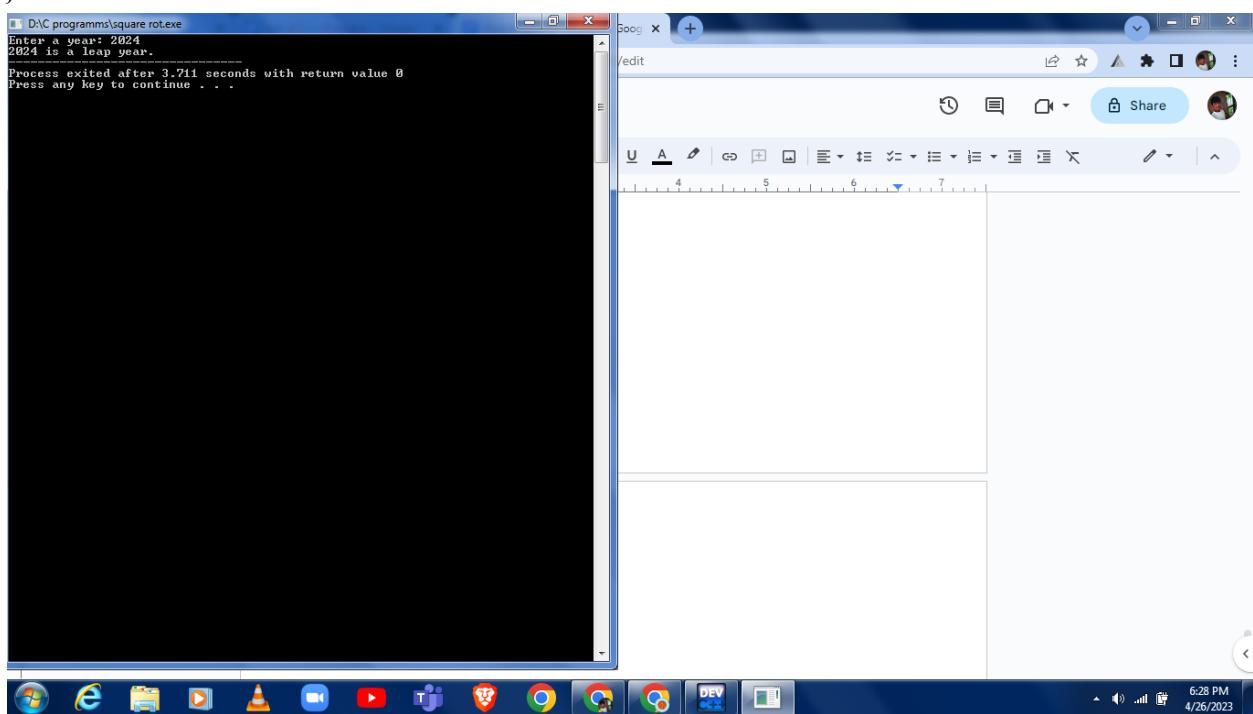
```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    double number,squareRoot;
    cout << "Enter a number: ";
    cin >> number;
    squareRoot = sqrt(number);
    cout << "The square root of " << number << " is " << squareRoot << endl;
    return 0;
}
```



14. leap year or not

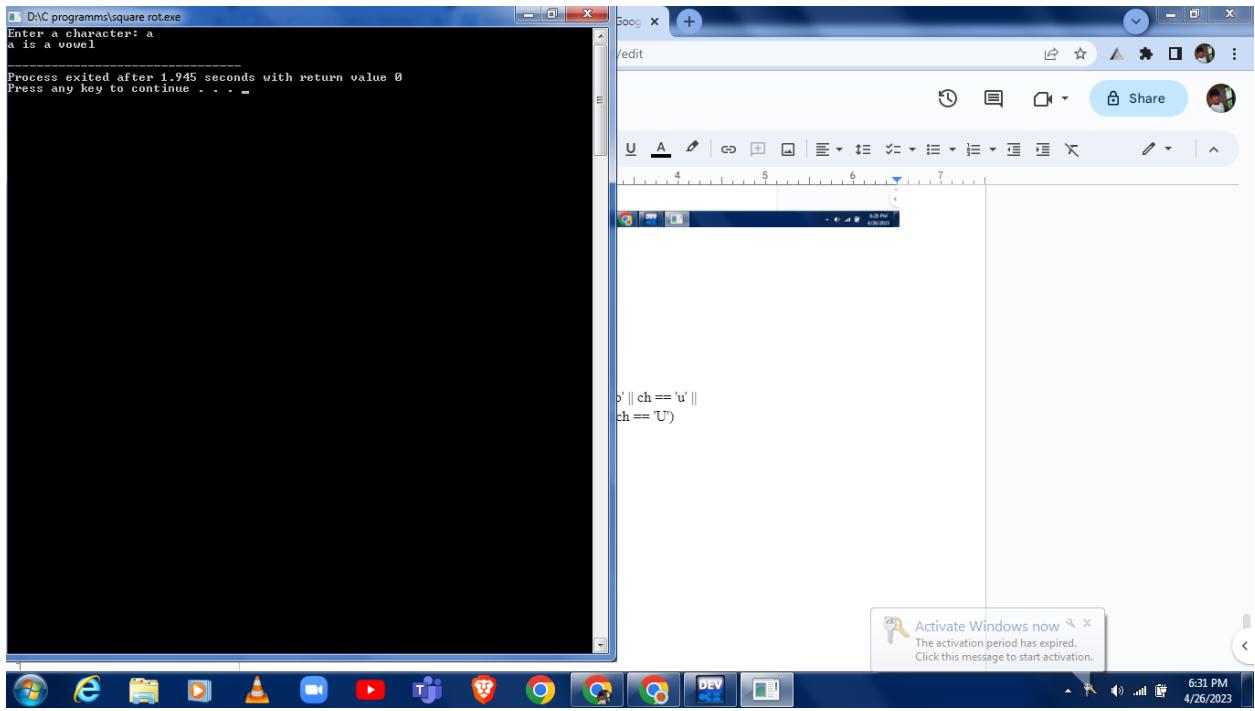
```
#include <iostream>
using namespace std;
int main()
{
    int year;
    cout << "Enter a year: ";
    cin >> year;
    if (year % 4 == 0)
    {
        if (year % 100 == 0)
        {
            if (year % 400 == 0)
                cout << year << " is a leap year.";
            else
                cout << year << " is not a leap year.";
        }
        else
            cout << year << " is a leap year.";
    }
    else
        cout << year << " is not a leap year.";
    return 0;
```

```
}
```



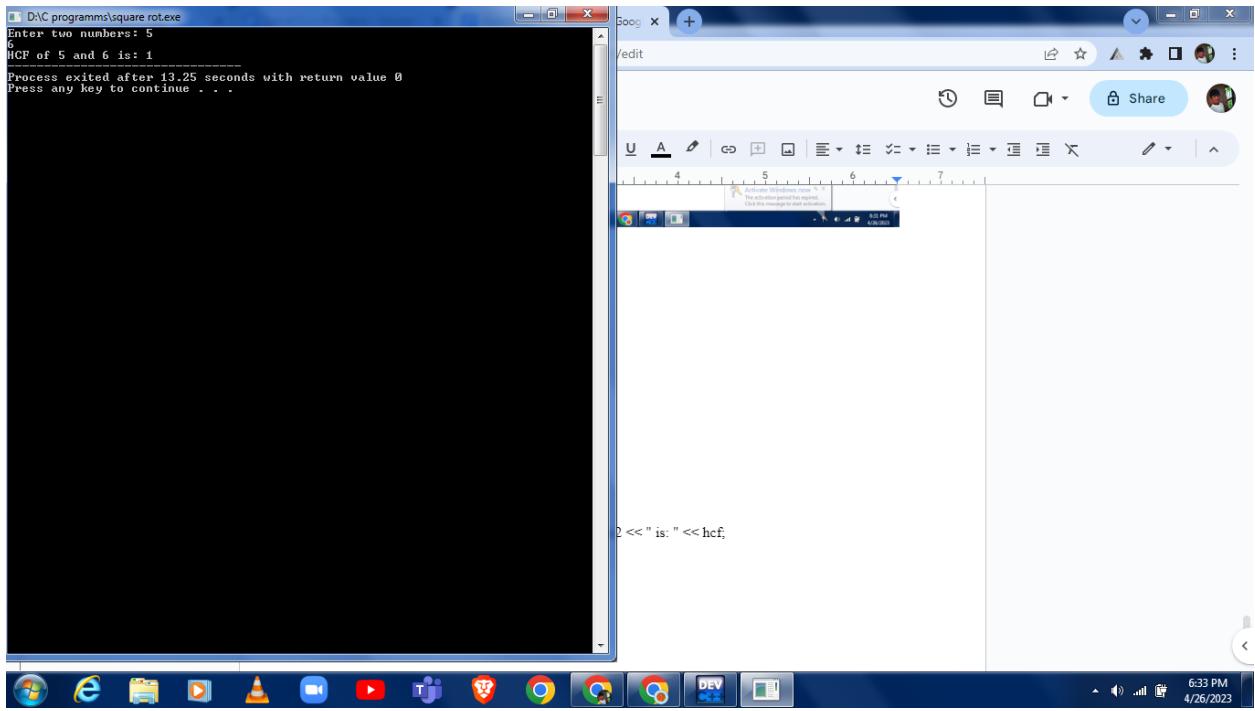
15.vowel or constant

```
#include <iostream>
using namespace std;
int main()
{
    char ch;
    cout << "Enter a character: ";
    cin >> ch;
    if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
        ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
    {
        cout << ch << " is a vowel" << endl;
    } else
    {
        cout << ch << " is a consonant" << endl;
    }
    return 0;
}
```



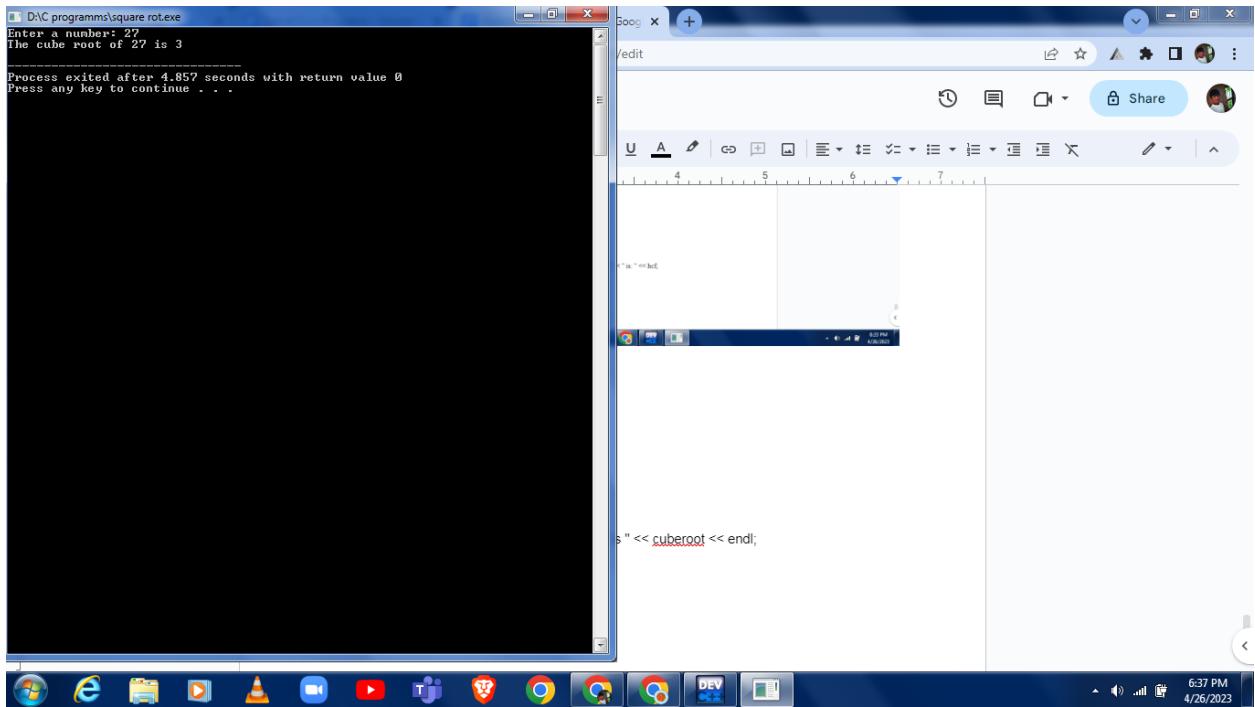
16.HCF

```
#include <iostream>
using namespace std;
int main()
{
    int num1,num2,hcf;
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;
    for(int i = 1; i <= num1 && i <= num2; i++)
    {
        if(num1 % i == 0 && num2 % i == 0)
        {
            hcf = i;
        }
    }
    cout << "HCF of " << num1 << " and " << num2 << " is: " << hcf;
    return 0;
}
```



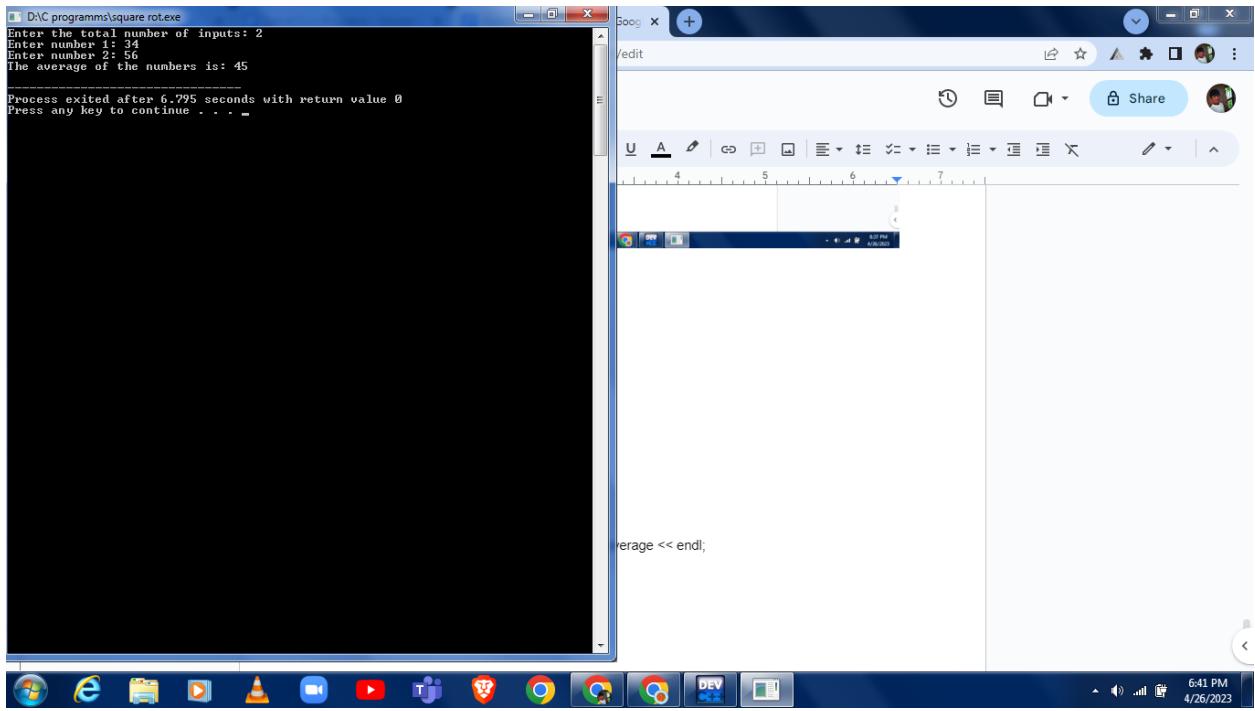
17. cube root

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    double num,cuberoot;
    cout << "Enter a number: ";
    cin >> num;
    cuberoot = cbrt(num);
    cout << "The cube root of " << num << " is " << cuberoot << endl;
    return 0;
}
```

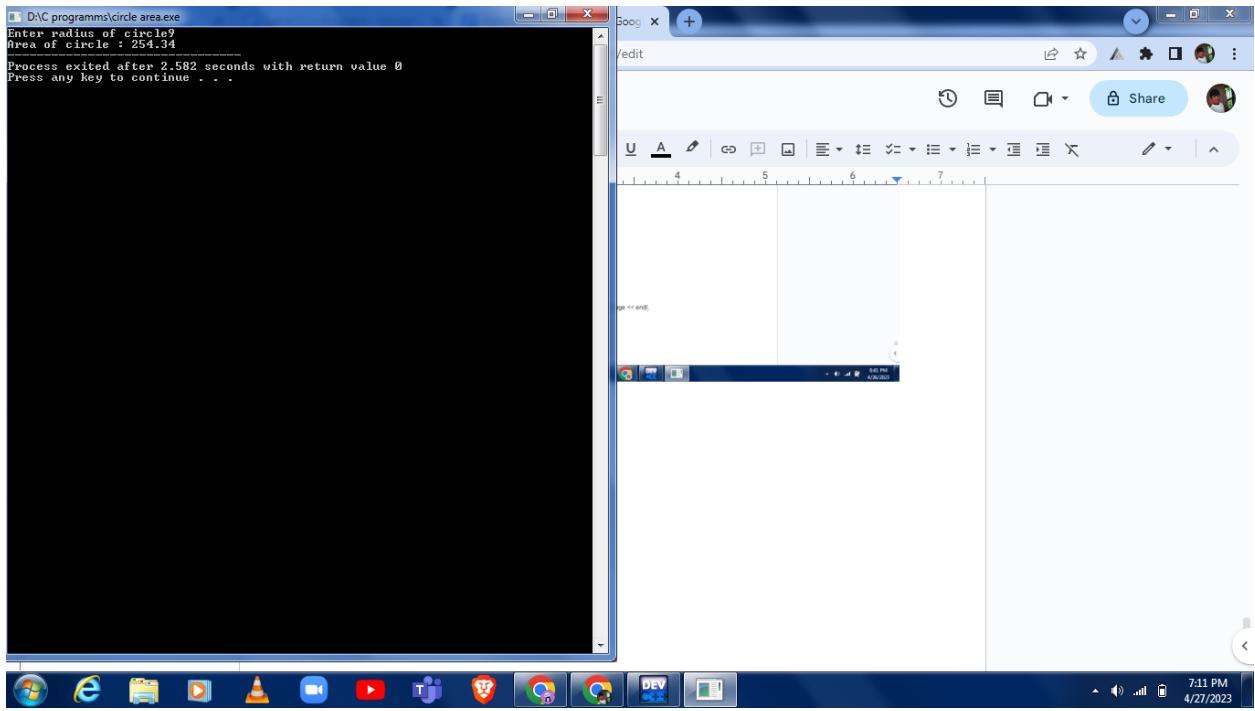


18. average of numbers

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    float num,sum,average;
    cout << "Enter the total number of inputs: ";
    cin >> n;
    for(int i = 1; i <= n; i++)
    {
        cout << "Enter number " << i << ":";
        cin >> num;
        sum += num;
    }
    average = sum / n;
    cout << "The average of the numbers is: " << average << endl;
    return 0;
}
```

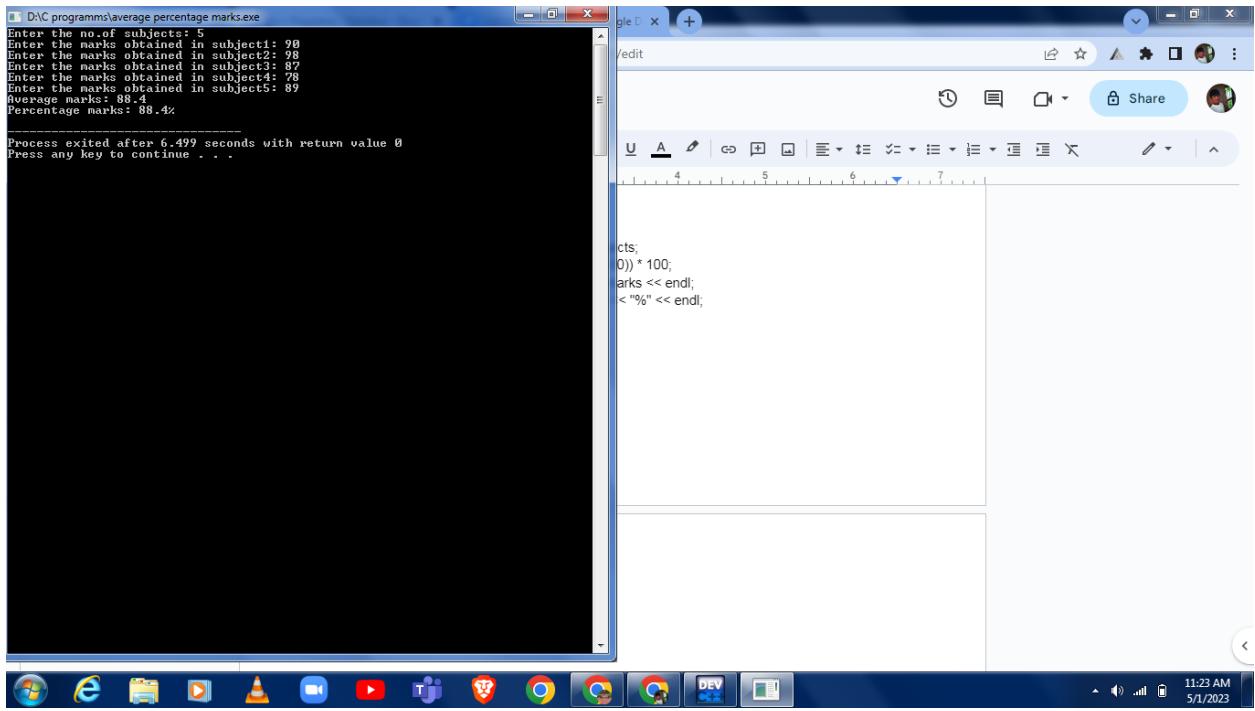


```
19.#include <iostream>
using namespace std;
int main()
{
    float radius,area,pi=3.14;
    cout << "Enter radius of circle ";
    cin >> radius;
    area = pi*radius*radius;
    cout << "Area of circle : " << area;
    return 0;
}
```



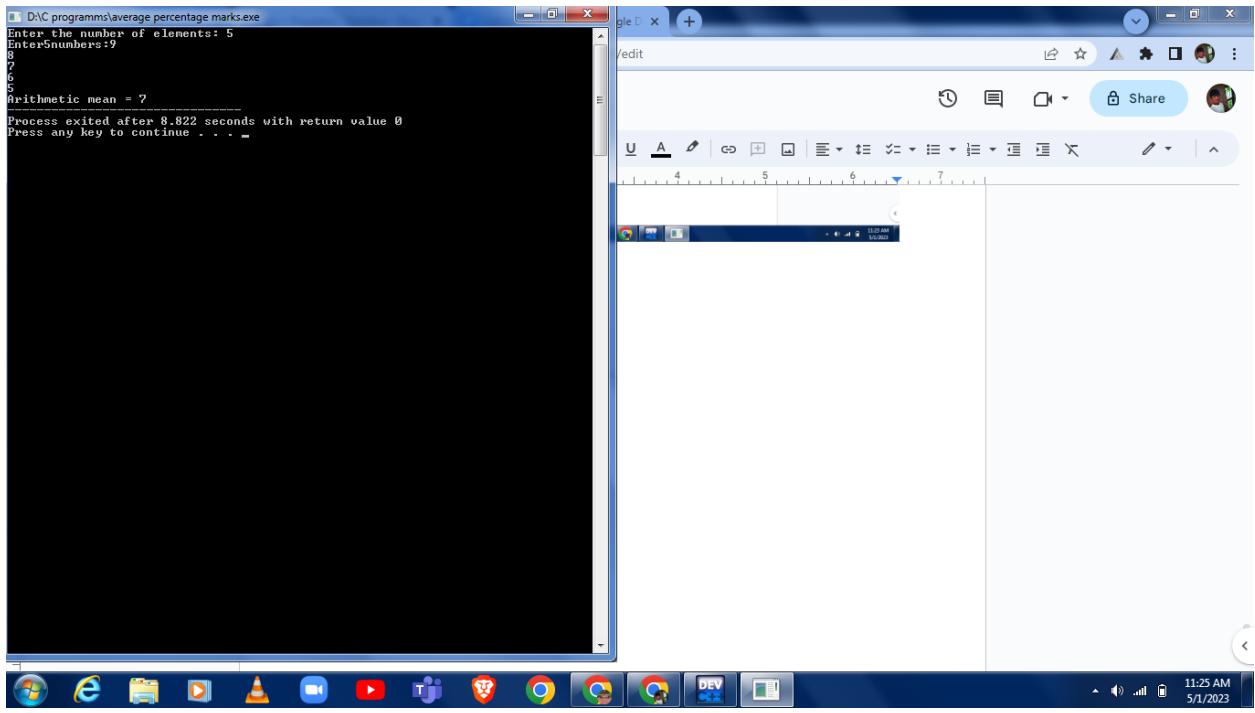
20. average percentage

```
#include <iostream>
using namespace std;
int main()
{
    int num_subjects;
    float total_marks,percentage,average_marks;
    cout << "Enter the no.of subjects: ";
    cin >> num_subjects;
    float marks[num_subjects];
    for (int i=0;i<num_subjects;i++)
    {
        cout << "Enter the marks obtained in subject" << i+1 << ": ";
        cin >> marks[i];
        total_marks += marks[i];
    }
    average_marks = total_marks/num_subjects;
    percentage = (total_marks/(num_subjects * 100)) * 100;
    cout << "Average marks: " << average_marks << endl;
    cout << "Percentage marks: " << percentage << "%" << endl;
    return 0;
}
```



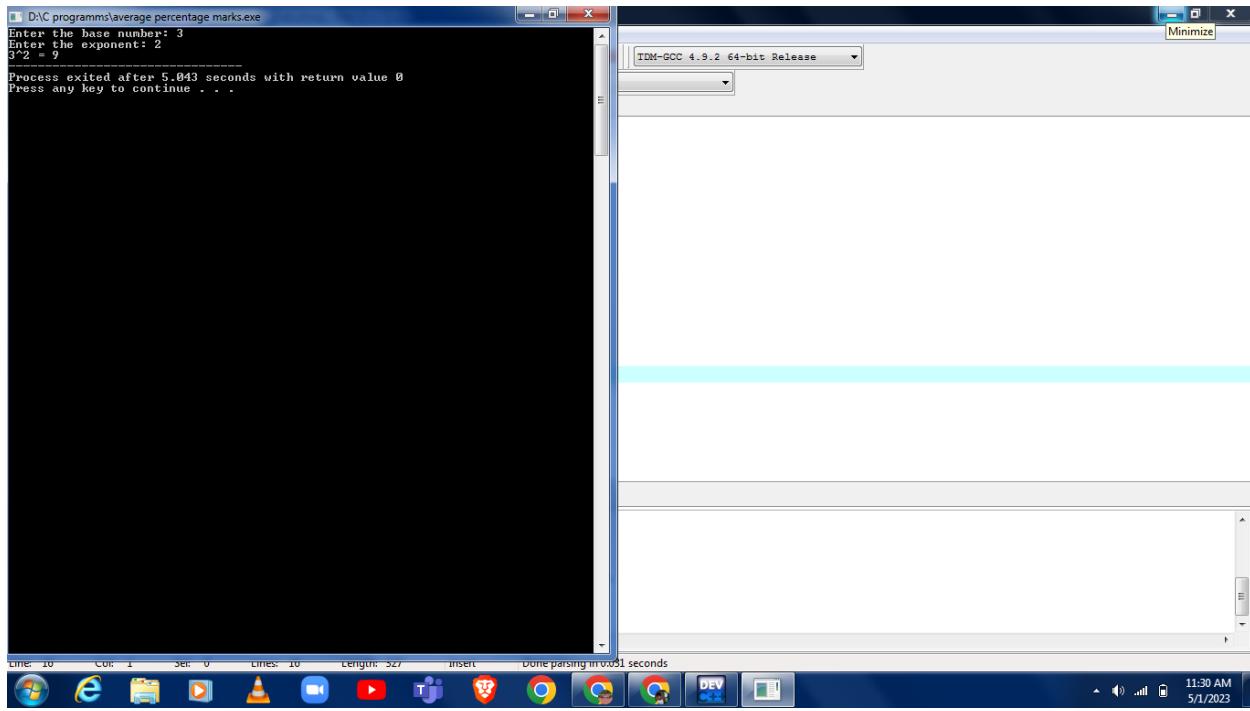
21.arithmetic mean

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    float sum,num;
    cout << "Enter the number of elements: ";
    cin >> n;
    cout << "Enter" << n << "numbers:";
    for (int i = 0; i < n; i++)
    {
        cin >> num;
        sum += num;
    }
    float mean = sum / n;
    cout << "Arithmetic mean = " << mean;
    return 0;
}
```



22.power of a number

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
    double base, exponent, result;
    cout << "Enter the base number: ";
    cin >> base;
    cout << "Enter the exponent: ";
    cin >> exponent;
    result = pow(base,exponent);
    cout << base << "^" << exponent << " = " << result;
    return 0;
}
```



23.program to find greatest b/w 3 nos. by defining the functions inside class

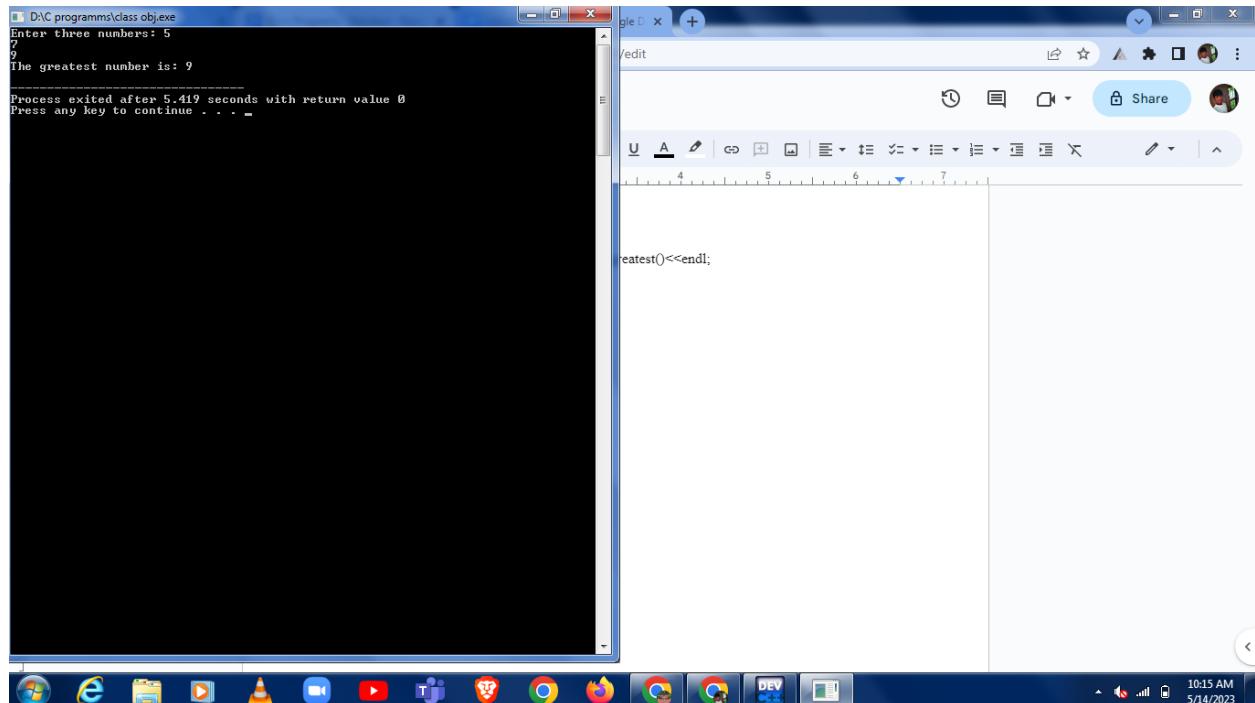
```
#include<iostream>
using namespace std;
class Greatest
{
    int num1,num2,num3;
public:
    void get_numbers(int a,int b,int c)
    {
        num1 = a;
        num2 = b;
        num3 = c;
    }
    int find_greatest()
    {
        int greatest = num1;
        if (num2 > greatest)
            greatest = num2;
        if (num3 > greatest)
            greatest = num3;
        return greatest;
    }
};
```

int main()

```

{
    Greatest obj;
    int num1, num2, num3;
    cout<<"Enter three numbers: ";
    cin>>num1>>num2>>num3;
    obj.get_numbers(num1, num2, num3);
    cout<<"The greatest number is: "<<obj.find_greatest()<<endl;
    return 0;
}

```

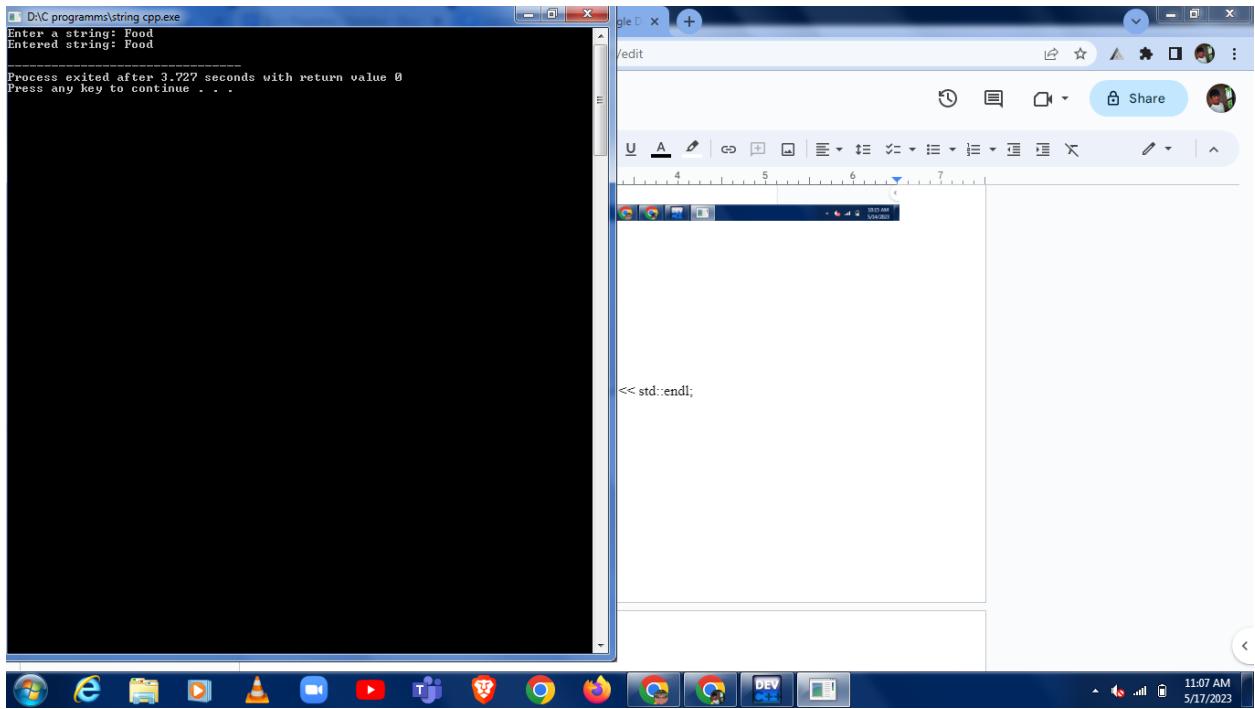


24.print entered string

```

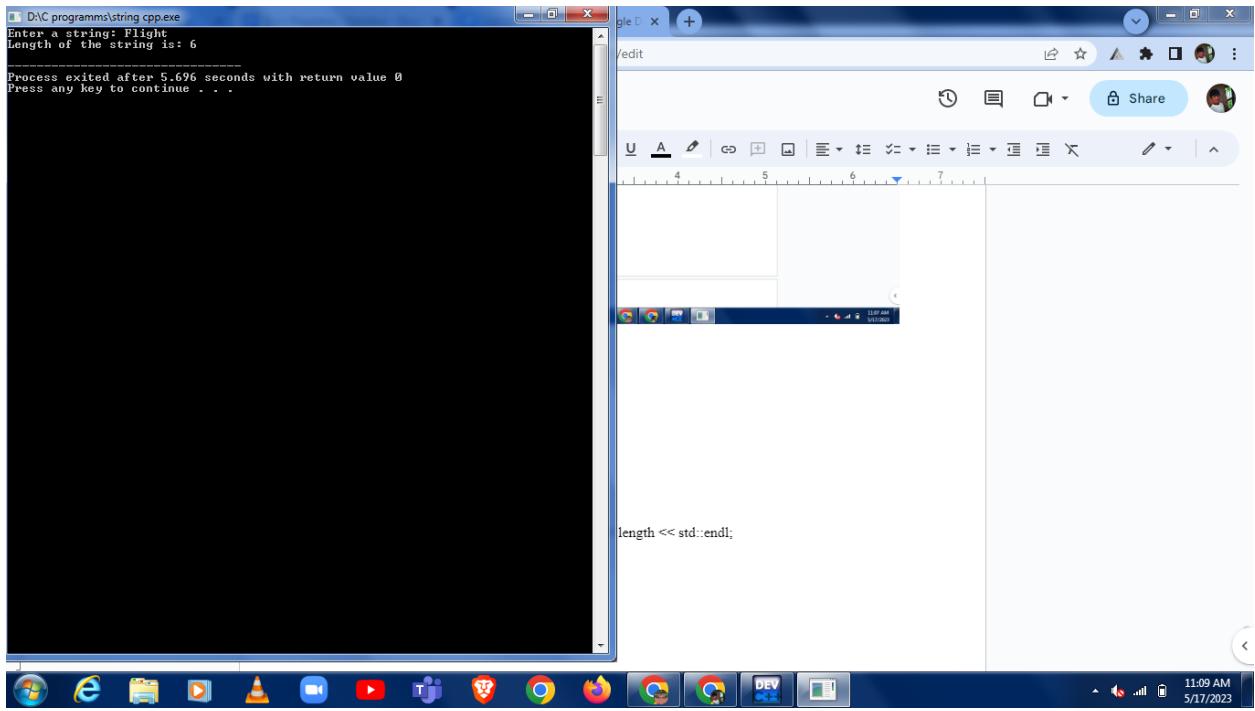
#include <iostream>
#include <string>
int main()
{
    std::string input;
    std::cout << "Enter a string: ";
    std::getline(std::cin, input);
    std::cout << "Entered string is: " << input << std::endl;
    return 0;
}

```



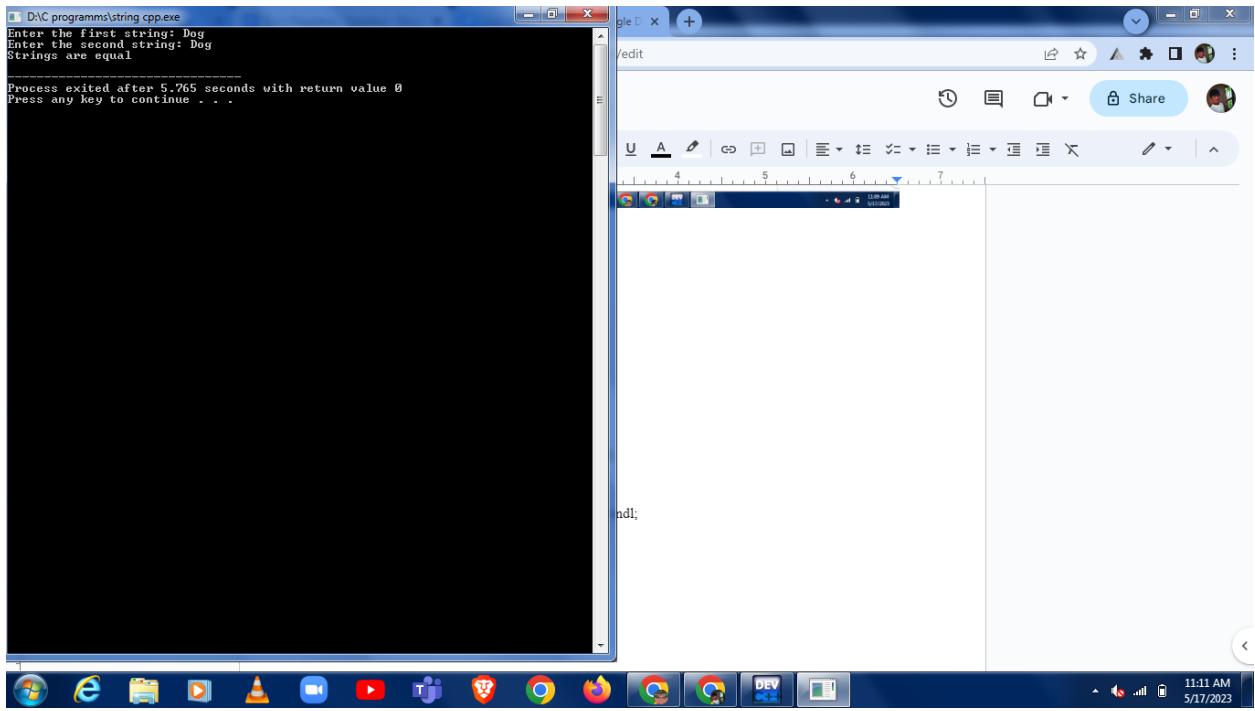
25.length of a string

```
#include <iostream>
#include <string>
int main()
{
    std::string str;
    std::cout << "Enter a string: ";
    std::getline(std::cin,str);
    int length = str.length();
    std::cout << "Length of the string is: " << length << std::endl;
    return 0;
}
```



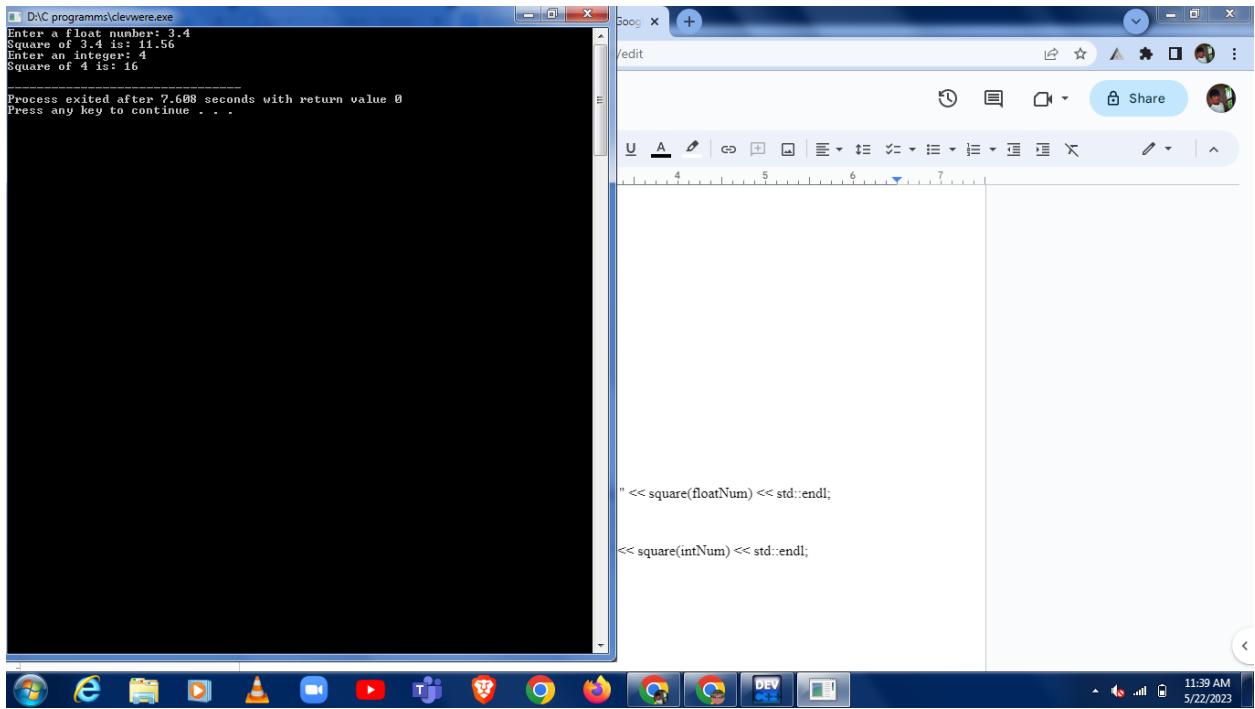
26.compare strings

```
#include <iostream>
#include <string>
int main()
{
    std::string str1,str2;
    std::cout << "Enter the first string: ";
    std::getline(std::cin,str1);
    std::cout << "Enter the second string: ";
    std::getline(std::cin,str2);
    if (str1 == str2)
    {
        std::cout << "Strings are equal" <<std::endl;
    } else
    {
        std::cout << "Strings are not equal" <<std::endl;
    }
    return 0;
}
```



27. square of float and integer using inline func

```
#include <iostream>
inline float square(float num)
{
    return num * num;
}
inline int square(int num)
{
    return num * num;
}
int main()
{
    float floatNum;
    int intNum;
    std::cout << "Enter a float number: ";
    std::cin >> floatNum;
    std::cout << "Square of " << floatNum << " is: " << square(floatNum) << std::endl;
    std::cout << "Enter an integer: ";
    std::cin >> intNum;
    std::cout << "Square of " << intNum << " is: " << square(intNum) << std::endl;
    return 0;
}
```



28. Program to display entered Date

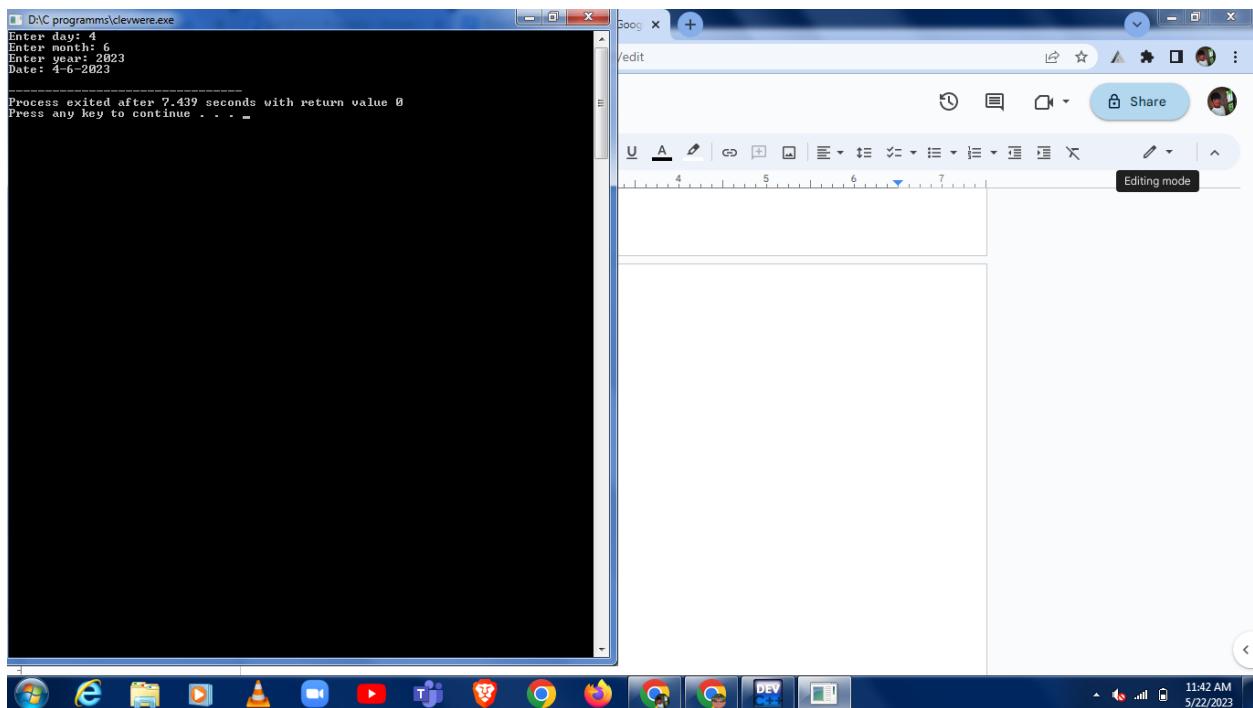
```
#include <iostream>
using namespace std;
class Date
{
private:
    int day;
    int month;
    int year;
public:
    void setDate(int d, int m, int y)
    {
        day = d;
        month = m;
        year = y;
    }
    void displayDate()
    {
        cout << "Date: " << day << "-" << month << "-" << year << endl;
    }
};
```

```
int main()
{
```

```

int day, month, year;
cout << "Enter day: ";
cin >> day;
cout << "Enter month: ";
cin >> month;
cout << "Enter year: ";
cin >> year;
Date date;
date.setDate(day, month, year);
date.displayDate();
return 0;
}

```



30. program to display Student details using class

```

#include <iostream>
#include <string>
using namespace std;
class Student
{
public:
    string name;
    int age;
    string rollNumber;
    void displayDetails()

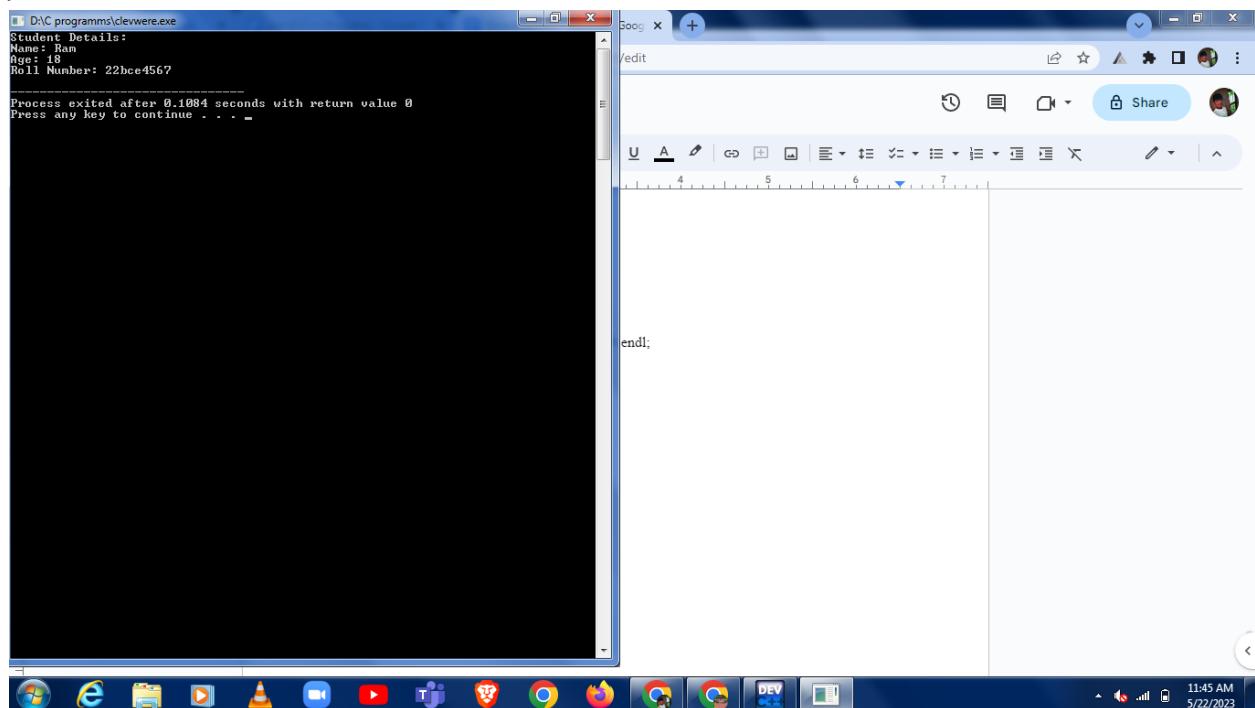
```

```

    {
        cout << "Name: " << name << endl;
        cout << "Age: " << age << endl;
        cout << "Roll Number: " << rollNumber << endl;
    }
};

int main()
{
    Student student1;
    student1.name = "Ram";
    student1.age = 18;
    student1.rollNumber = "22bce4567";
    cout << "Student Details:" << endl;
    student1.displayDetails();
    return 0;
}

```



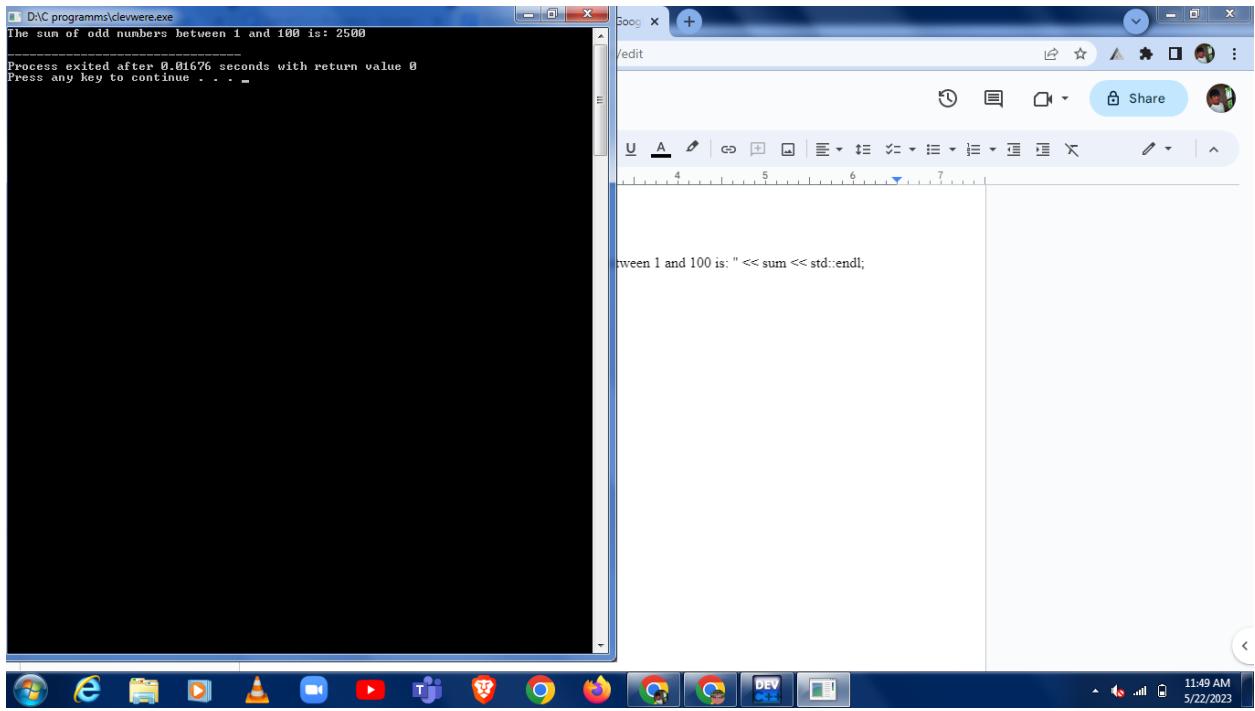
31. Program to find Sum of odd numbers between 1 and 100 using class

```

#include <iostream>
class OddSum
{
private:
    int start;
    int end;

```

```
public:  
    OddSum(int start, int end)  
    {  
        this->start = start;  
        this->end = end;  
    }  
    int Sum()  
    {  
        int sum = 0;  
        for (int num = start; num <= end; num++)  
        {  
            if (num % 2 != 0)  
            {  
                sum += num;  
            }  
        }  
        return sum;  
    }  
};  
int main() {  
    OddSum calculator(1,100);  
    int sum = calculator.Sum();  
    std::cout << "The sum of odd numbers between 1 and 100 is: " << sum << std::endl;  
    return 0;  
}
```



31. program to display Entered Time using class

```
#include <iostream>
using namespace std;
class Time
{
private:
    int hours;
    int minutes;
    int seconds;
public:
    void setTime(int h, int m, int s)
    {
        hours = h;
        minutes = m;
        seconds = s;
    }

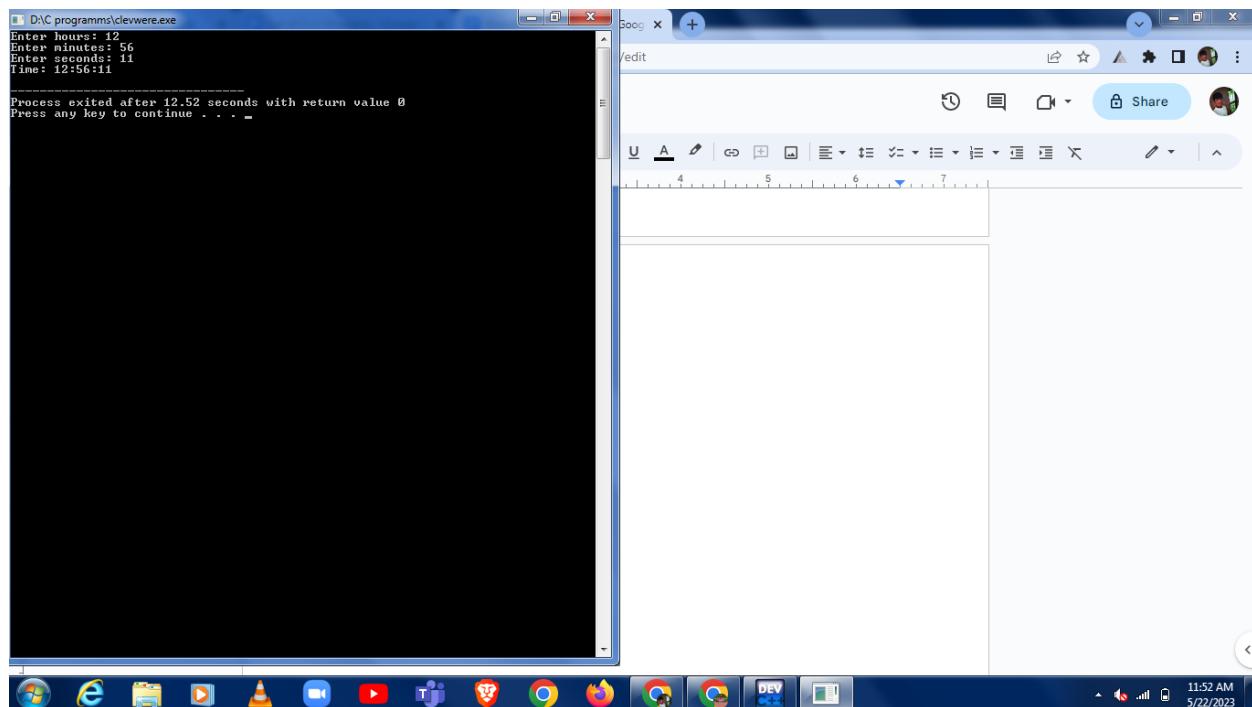
    void displayTime()
    {
        cout << "Time: " << hours << ":" << minutes << ":" << seconds << endl;
    }
};

int main()
```

```

{
    int h, m, s;
    cout << "Enter hours: ";
    cin >> h;
    cout << "Enter minutes: ";
    cin >> m;
    cout << "Enter seconds: ";
    cin >> s;
    Time t;
    t.setTime(h,m,s);
    t.displayTime();
    return 0;
}

```



32. program to find Largest among 2 numbers using class

```

#include <iostream>
class Number
{
    private:
    int num1, num2;
    public:
    Number(int a, int b)
    {
        num1 = a;

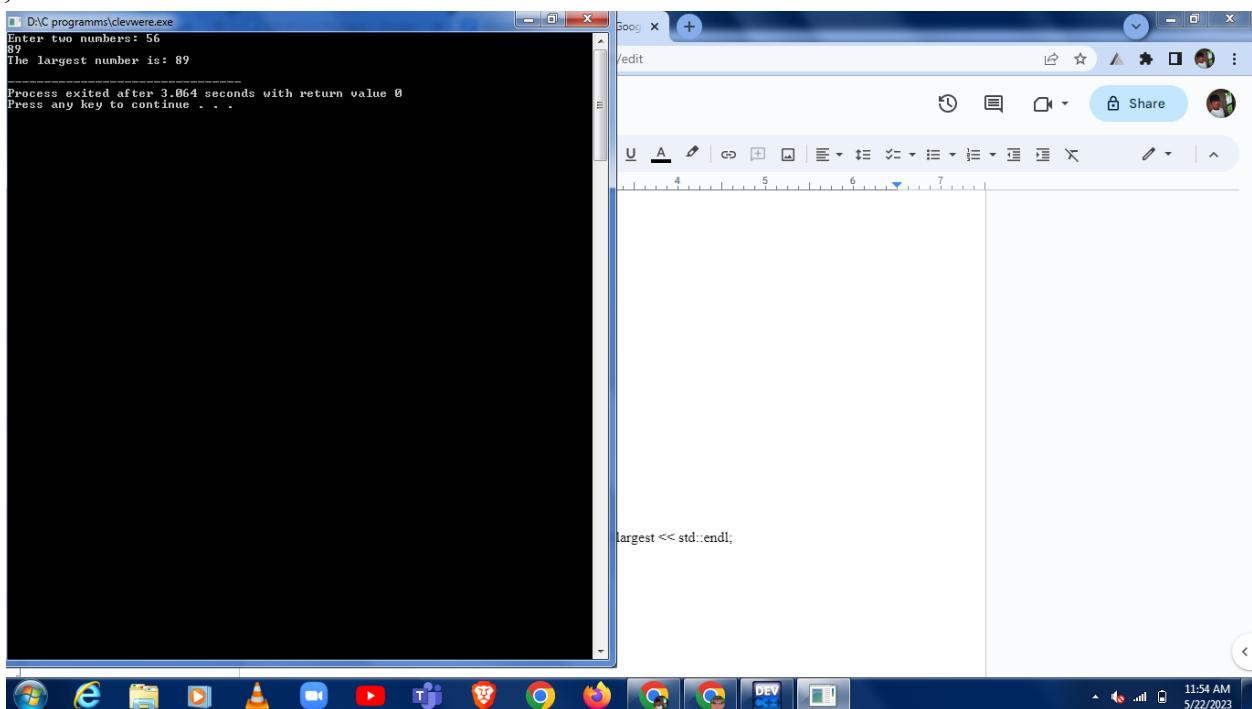
```

```

        num2 = b;
    }
    int findLargest()
    {
        if (num1 > num2)
        {
            return num1;
        } else
        {
            return num2;
        }
    }
};

int main() {
    int a,b;
    std::cout << "Enter two numbers: ";
    std::cin >> a >> b;
    Number num(a,b);
    int largest = num.findLargest();
    std::cout << "The largest number is: " << largest << std::endl;
    return 0;
}

```



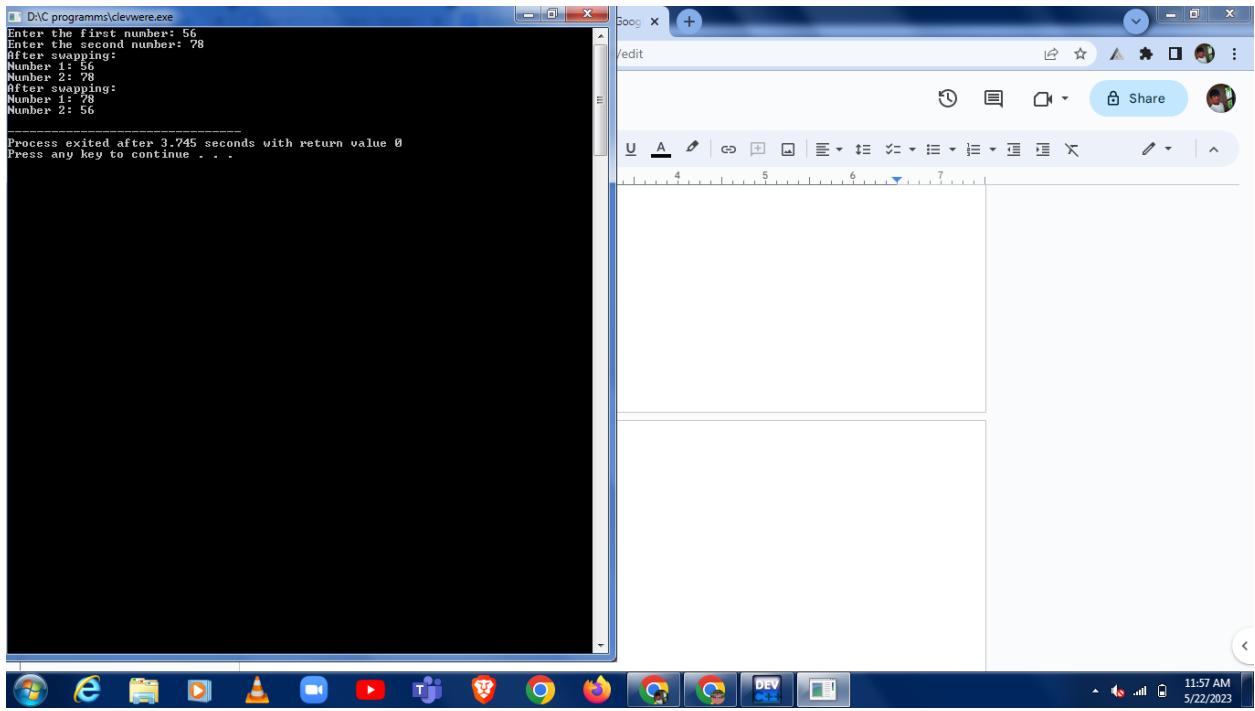
33.swap two num using class

```
#include <iostream>
using namespace std;
class NumberSwap
{
private:
    int num1, num2;
public:
    NumberSwap(int n1, int n2)
    {
        num1 = n1;
        num2 = n2;
    }

    void swapNumbers()
    {
        int temp = num1;
        num1 = num2;
        num2 = temp;
    }

    void displayNumbers() {
        cout << "After swapping: " << endl;
        cout << "Number 1: " << num1 << endl;
        cout << "Number 2: " << num2 << endl;
    }
};

int main()
{
    int n1, n2;
    cout << "Enter the first number: ";
    cin >> n1;
    cout << "Enter the second number: ";
    cin >> n2;
    NumberSwap swap(n1, n2);
    swap.displayNumbers();
    swap.swapNumbers();
    swap.displayNumbers();
    return 0;
}
```



34. Program to calculate Volume of Cube using constructor and destructor

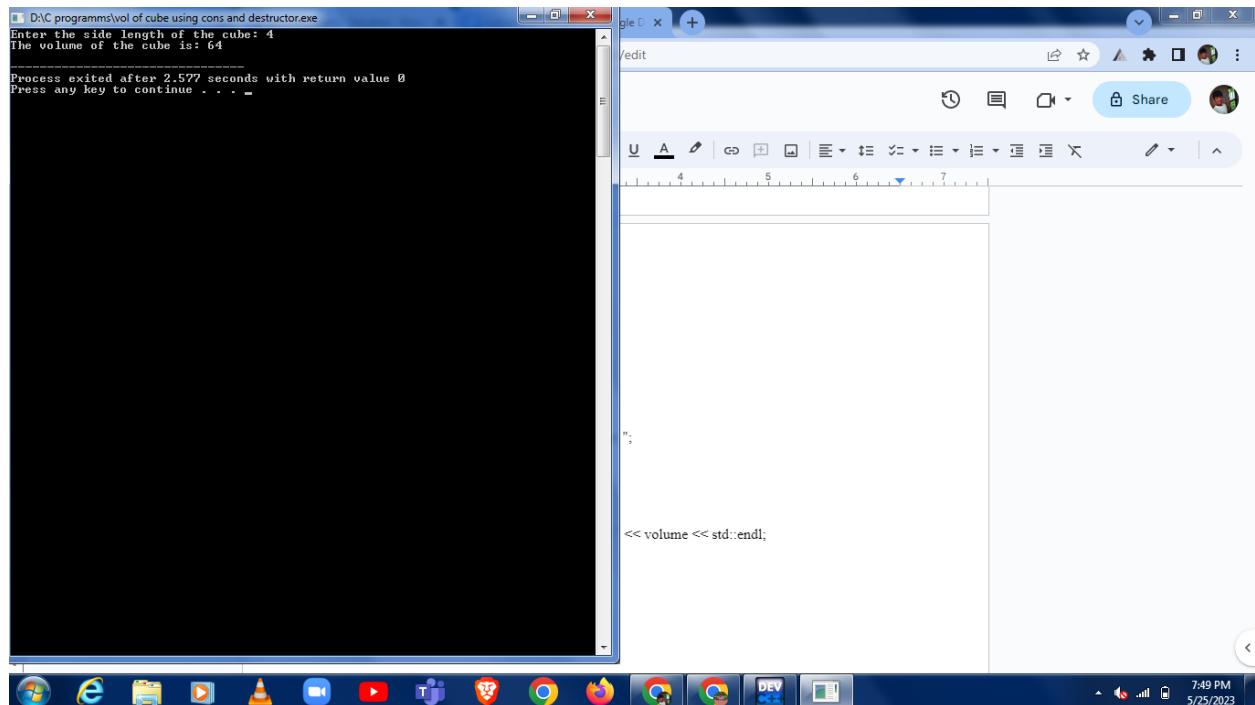
```
#include <iostream>
class Cube
{
private:
    double side;
    double volume;
public:
    Cube(double side)
    {
        this->side = side;
        this->volume = 0.0;
    }
    Cube()
    {
        std::cout << " destroyed cube object ." << std::endl;
    }
    void calculateVolume()
    {
        volume = side * side * side;
    }
    double getVolume()
    {
```

```

        return volume;
    }
};

int main()
{
    double sideLength;
    std::cout << "Enter the side length of the cube: ";
    std::cin >> sideLength;
    Cube cube(sideLength);
    cube.calculateVolume();
    double volume = cube.getVolume();
    std::cout << "The volume of the cube is: " << volume << std::endl;
    return 0;
}

```



35. Program to Find Even & Odd Elements in Array

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

```
void findEvenOdd(int arr[], int size) {
    cout << "Even elements: ";
    for (int i = 0; i < size; i++) {
        if (arr[i] % 2 == 0) {
            cout << arr[i] << " ";
```

```
        }
    }

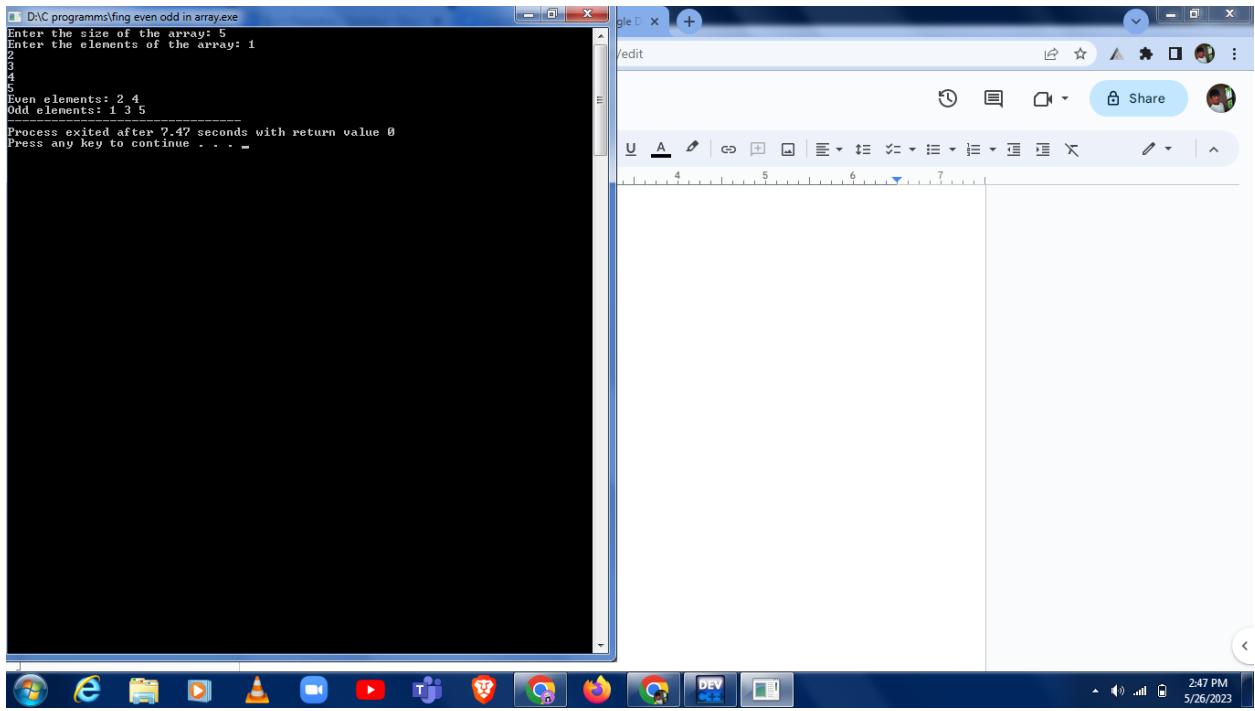
cout << "\nOdd elements: ";
for (int i = 0; i < size; i++) {
    if (arr[i] % 2 != 0) {
        cout << arr[i] << " ";
    }
}
}

int main() {
    int size;
    cout << "Enter the size of the array: ";
    cin >> size;

    int arr[size];
    cout << "Enter the elements of the array: ";
    for (int i = 0; i < size; i++) {
        cin >> arr[i];
    }

    findEvenOdd(arr, size);

    return 0;
}
```



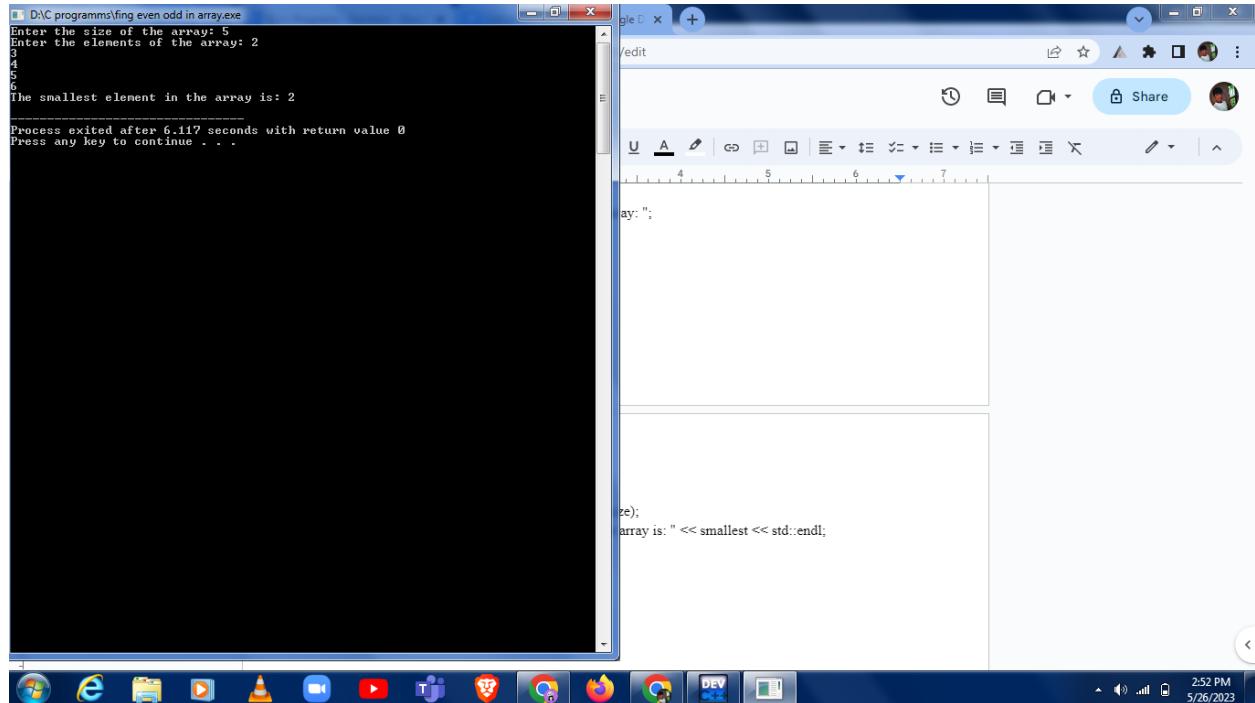
36.smallest num in a array

```
#include <iostream>
int findSmallestElement(int arr[], int size)
{
    int smallest = arr[0];
    for (int i = 1; i < size; i++) {
        if (arr[i] < smallest) {
            smallest = arr[i];
        }
    }
    return smallest;
}
int main()
{
    int size;
    std::cout << "Enter the size of the array: ";
    std::cin >> size;
    int arr[size];
    std::cout << "Enter the elements of the array: ";
    for (int i = 0; i < size; i++)
    {
        std::cin >> arr[i];
    }
```

```

        int smallest = findSmallestElement(arr, size);
        std::cout << "The smallest element in the array is: " << smallest << std::endl;
        return 0;
    }
}

```



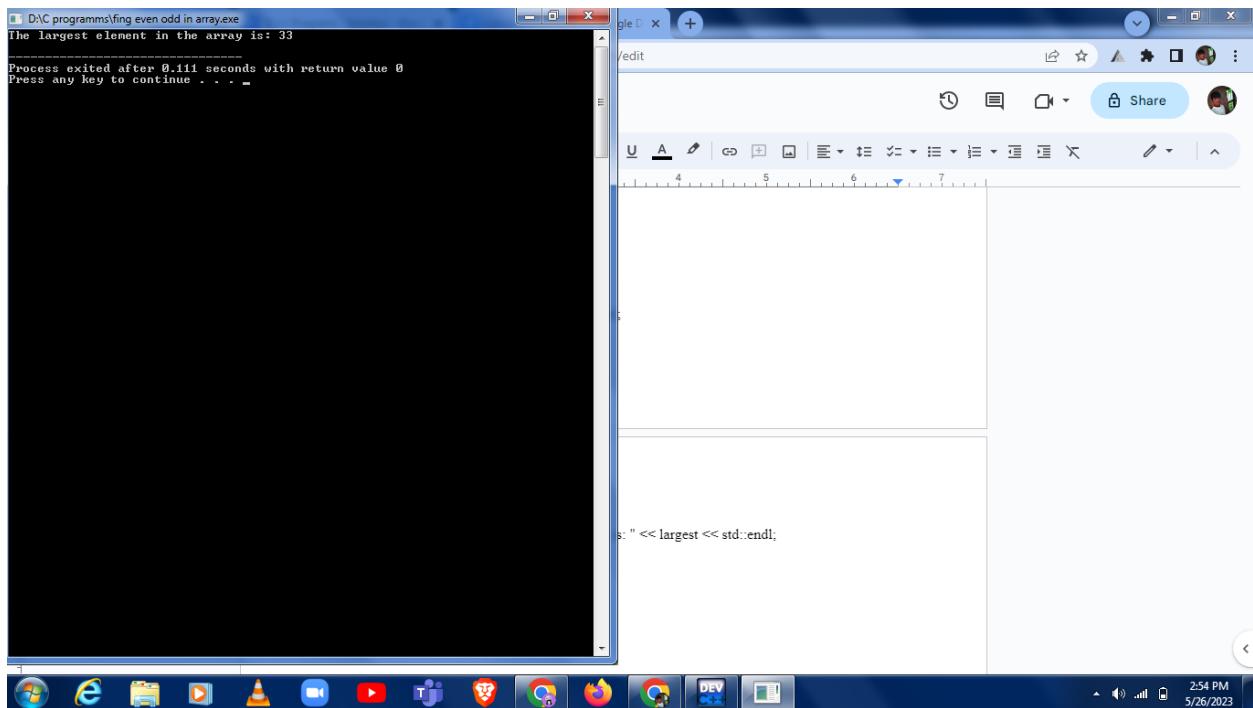
37. Program to Find Largest Element in Array

```

#include <iostream>
int findLargestElement(int arr[], int size)
{
    int max = arr[0];
    for (int i = 1; i < size; i++)
    {
        if (arr[i] > max)
        {
            max = arr[i];
        }
    }
    return max;
}
int main()
{
    int arr[] = {10, 25, 8, 16, 33};
    int size = sizeof(arr) / sizeof(arr[0]);
    int largest = findLargestElement(arr, size);
}

```

```
    std::cout << "The largest element in the array is: " << largest << std::endl;
    return 0;
}
```



38.10 multiplier

```
#include <iostream>
class Multiply
{
private:
    int* arr;
    int size;
public:
    Multiply(int* array, int length)
    {
        arr = array;
        size = length;
    }
    void multiplyBy10()
    {
        for (int i = 0; i < size; i++)
        {
            arr[i] *= 10;
        }
    }
}
```

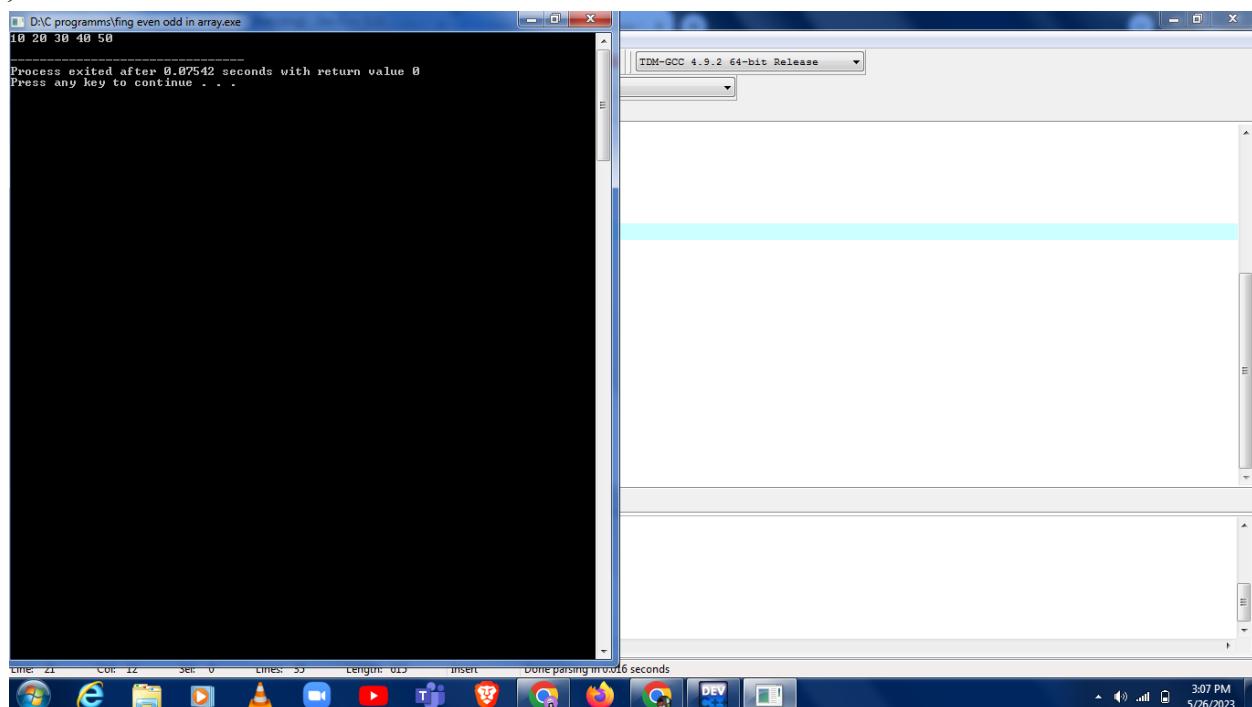
```

};

int main()
{
    int array[] = {1, 2, 3, 4, 5};
    int length = sizeof(array) / sizeof(array[0]);

    Multiply multiply(array, length);
    multiply.multiplyBy10();
    for (int i = 0; i < length; i++)
    {
        std::cout << array[i] << " ";
    }
    std::cout << std::endl;
    return 0;
}

```



39.octal to binary

```

#include <iostream>
#include <string>
std::string octalToBinary(char octalDigit)
{
    std::string binaryDigit;

    switch (octalDigit)

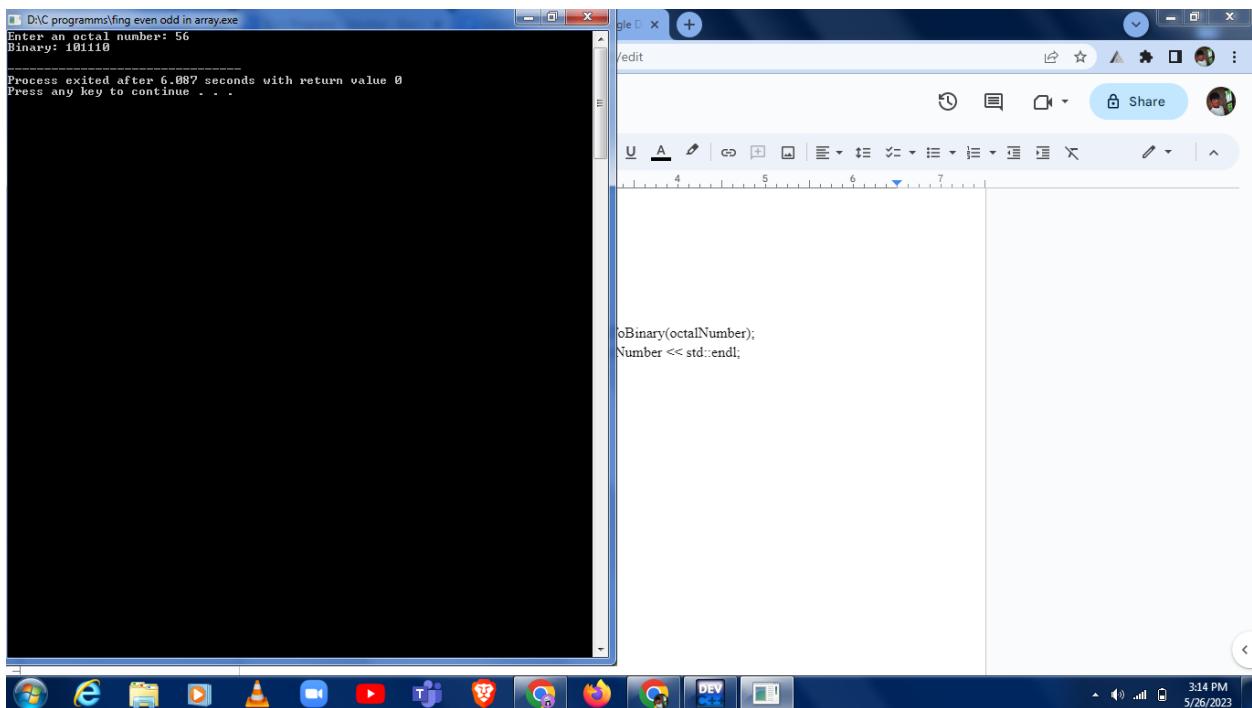
```

```
{  
    case '0':  
        binaryDigit = "000";  
        break;  
    case '1':  
        binaryDigit = "001";  
        break;  
    case '2':  
        binaryDigit = "010";  
        break;  
    case '3':  
        binaryDigit = "011";  
        break;  
    case '4':  
        binaryDigit = "100";  
        break;  
    case '5':  
        binaryDigit = "101";  
        break;  
    case '6':  
        binaryDigit = "110";  
        break;  
    case '7':  
        binaryDigit = "111";  
        break;  
    default:  
        binaryDigit = "000";  
        break;  
}  
  
return binaryDigit;  
}  
std::string convertOctalToBinary(const std::string& octalNumber)  
{  
    std::string binaryNumber;  
    for (char digit : octalNumber)  
    {  
        std::string binaryDigit = octalToBinary(digit);  
        binaryNumber += binaryDigit;  
    }  
}
```

```

        return binaryNumber;
    }
int main()
{
    std::string octalNumber;
    std::cout << "Enter an octal number: ";
    std::cin >> octalNumber;
    std::string binaryNumber = convertOctalToBinary(octalNumber);
    std::cout << "Binary number: " << binaryNumber << std::endl;
    return 0;
}

```



40.octal to decimal

```

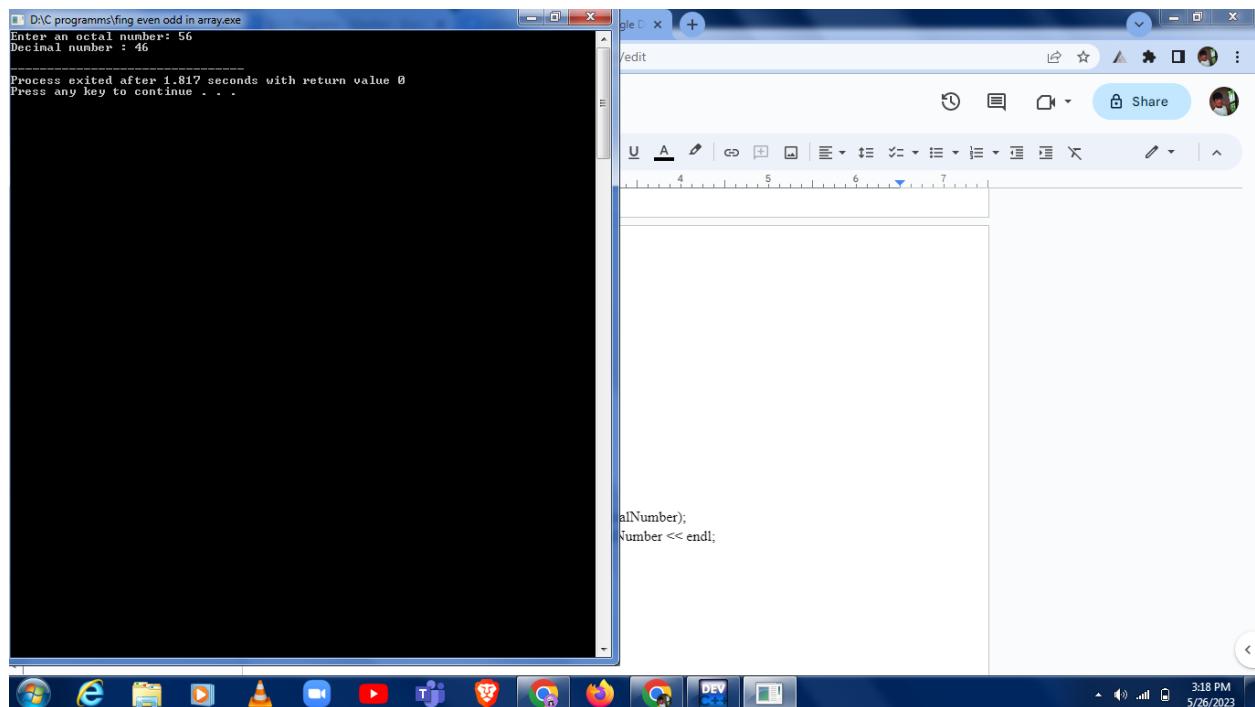
#include <iostream>
#include <cmath>
using namespace std;
int octalToDecimal(int octal)
{
    int decimal = 0;
    int i = 0;
    while (octal != 0)
    {
        int remainder = octal % 10;
        decimal += remainder * pow(8, i);

```

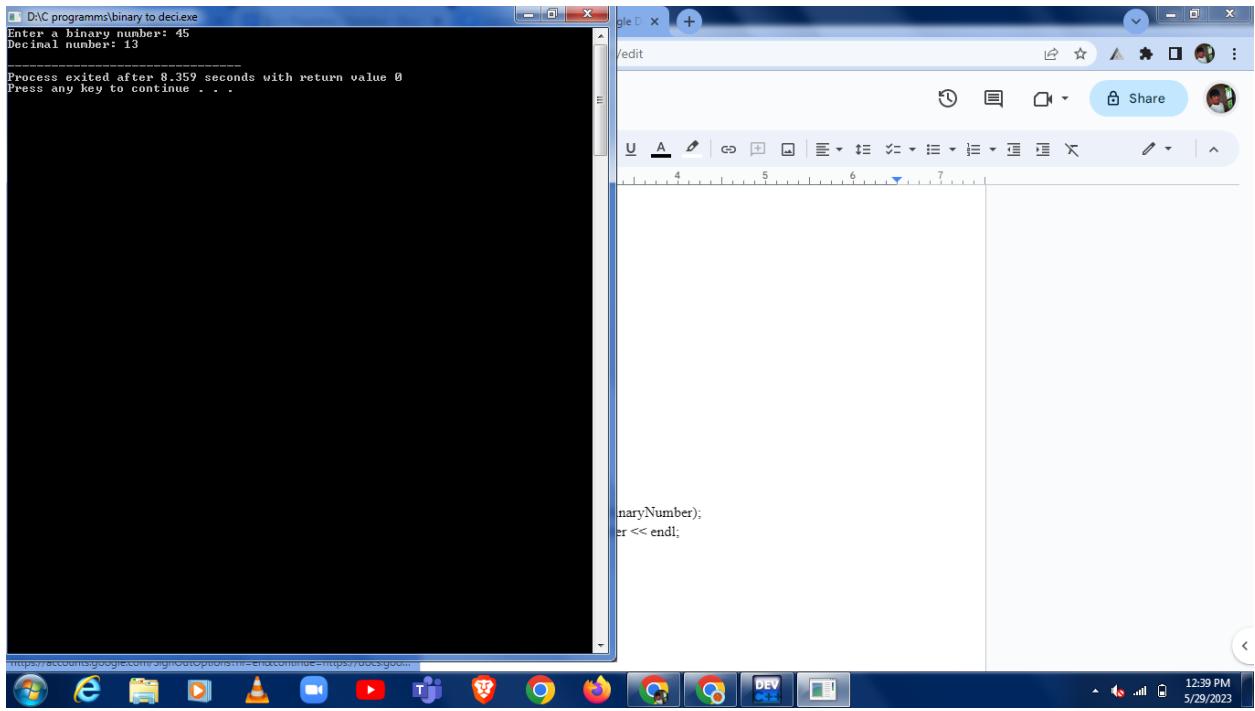
```

        ++i;
        octal /= 10;
    }
    return decimal;
}
int main()
{
    int octalNumber;
    cout << "Enter an octal number: ";
    cin >> octalNumber;
    int decimalNumber = octalToDecimal(octalNumber);
    cout << "Decimal number : " << decimalNumber << endl;
    return 0;
}

```

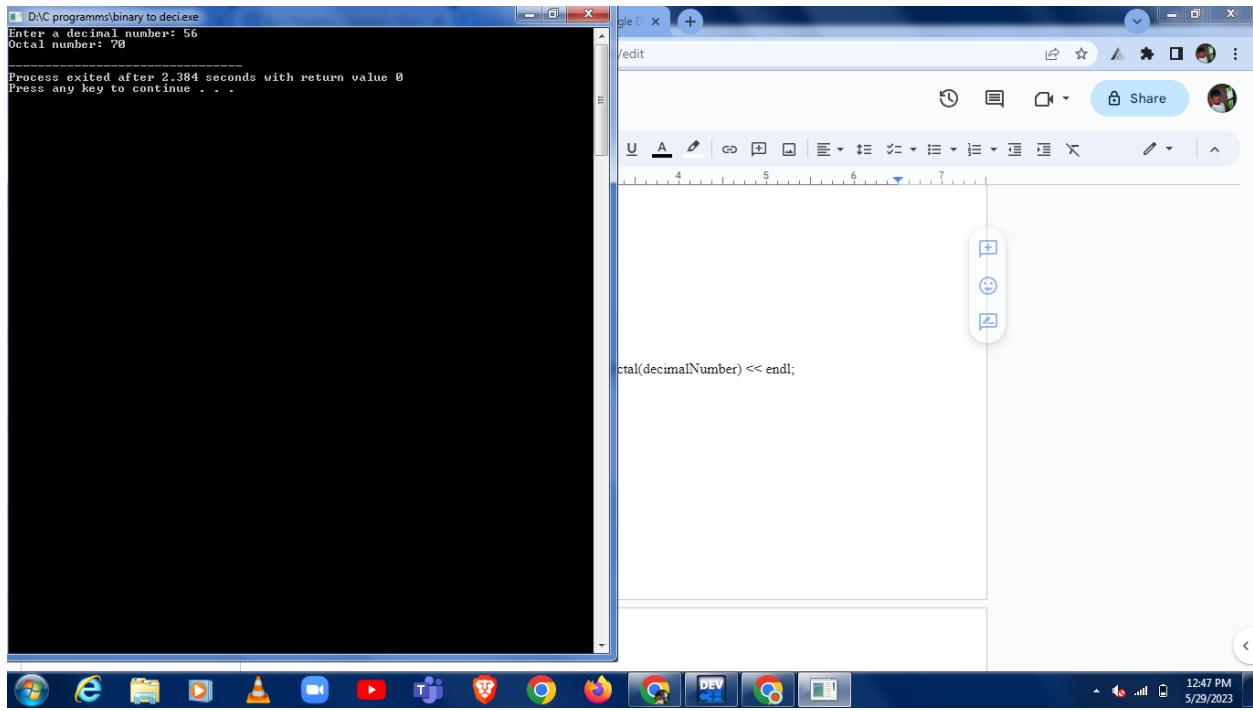


```
41.binary to deci
#include <iostream>
#include <cmath>
using namespace std;
int binaryToDecimal(int binary)
{
    int decimal = 0;
    int base = 1;while (binary > 0)
    {
        int lastDigit = binary % 10;
        decimal += lastDigit * base;
        binary /= 10;
        base *= 2;
    }
    return decimal;
}
int main()
{
    int binaryNumber;
    cout << "Enter a binary number: ";
    cin >> binaryNumber;
    int decimalNumber = binaryToDecimal(binaryNumber);
    cout << "Decimal number: " << decimalNumber << endl;
    return 0;
}
```



42. Program to Convert deci to octal

```
#include <iostream>
using namespace std;
int decimalToOctal(int decimalNumber) {
    int octalNumber = 0, i = 1;
    while (decimalNumber != 0) {
        octalNumber += (decimalNumber % 8) * i;
        decimalNumber /= 8;
        i *= 10;
    }
    return octalNumber;
}
int main() {
    int decimalNumber;
    cout << "Enter a decimal number: ";
    cin >> decimalNumber;
    cout << "Octal number: " << decimalToOctal(decimalNumber) << endl;
    return 0;
}
```



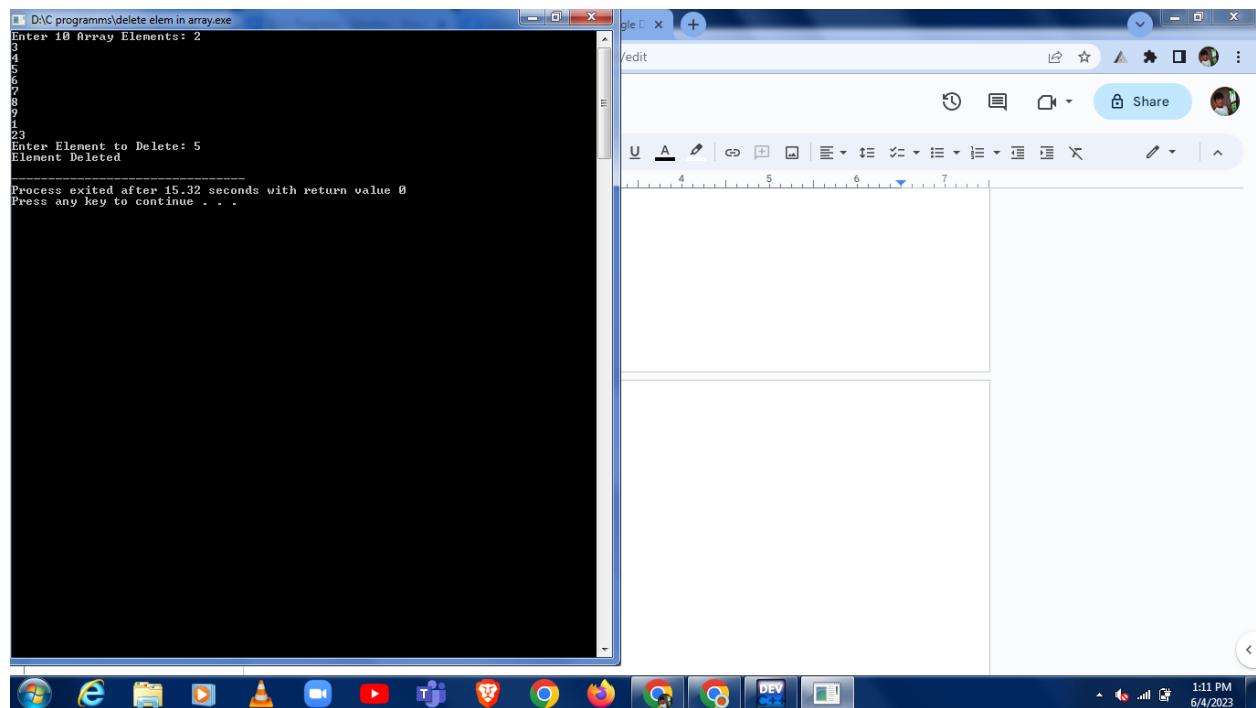
43.Program to Delete element from Array

```
#include<iostream>
using namespace std;
int main()
{
    int arr[10],total=10,i;element,j,answer;
    cout<<"Enter 10 Array Elements: ";
    for(i=0;i<total;i++)
        cin>>arr[i];
    cout<<"Enter Element to Delete: ";
    cin>>element;
    for(i=0;i<total;i++)
    {
        if(arr[i]==element)
        {
            for(j=i;j<(total-1);j++)
                arr[j] = arr[j+1];
            answer++;
            i--;
            total--;
        }
    }
}
```

```

if(answer==0)
    cout<<"Element not found in the array";
else
    cout<<"Element Deleted";
cout<<endl;
return 0;
}

```



44.student details constructor and destructor

```

#include <iostream>
#include <string>
class Student
{
private:
    std::string name;
    int age;
    std::string course;
public:
    Student(const std::string& n, int a, const std::string& c) : name(n), age(a), course(c)
    {
        std::cout << "Constructor called." << std::endl;
    }
    Student()

```

```

    {
        std::cout << "Destructor called." << std::endl;
    }

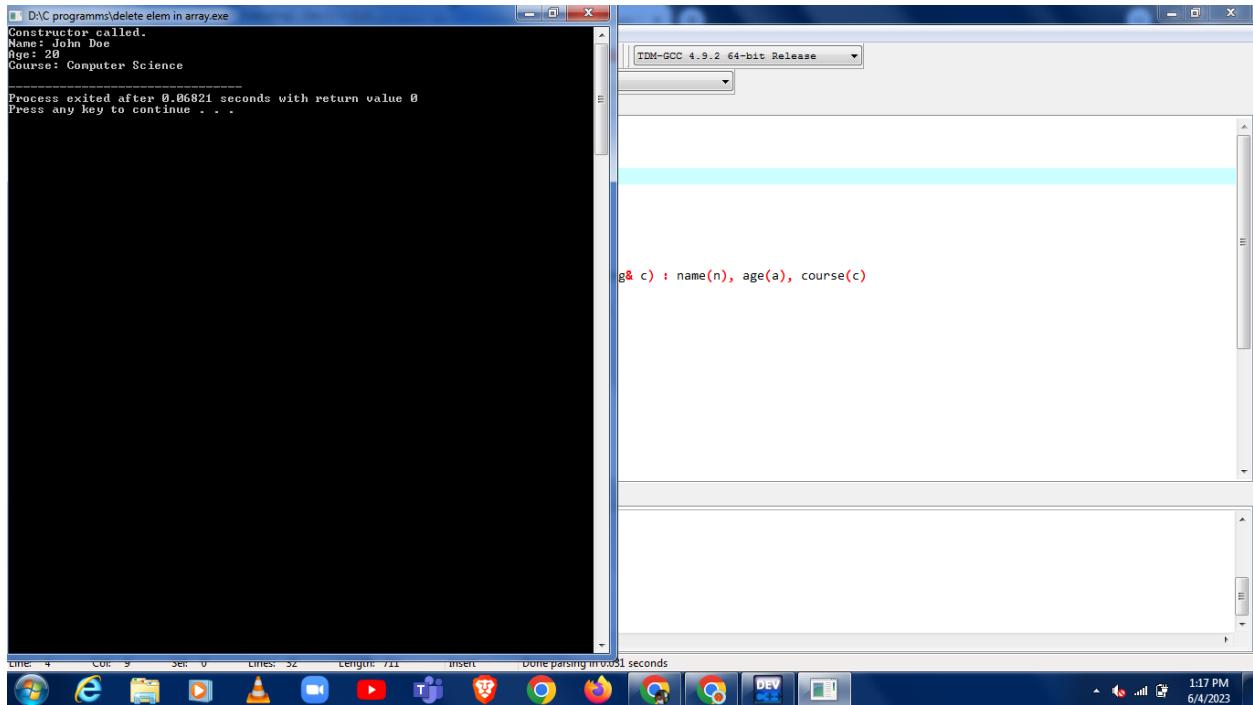
    void displayDetails()
    {
        std::cout << "Name: " << name << std::endl;
        std::cout << "Age: " << age << std::endl;
        std::cout << "Course: " << course << std::endl;
    }

};

int main()
{

```

Student student("John Doe",20,"Computer Science");
 student.displayDetails();
 return 0;

}


45.area of rect using constructor

```
#include <iostream>
```

```
class Rectangle
```

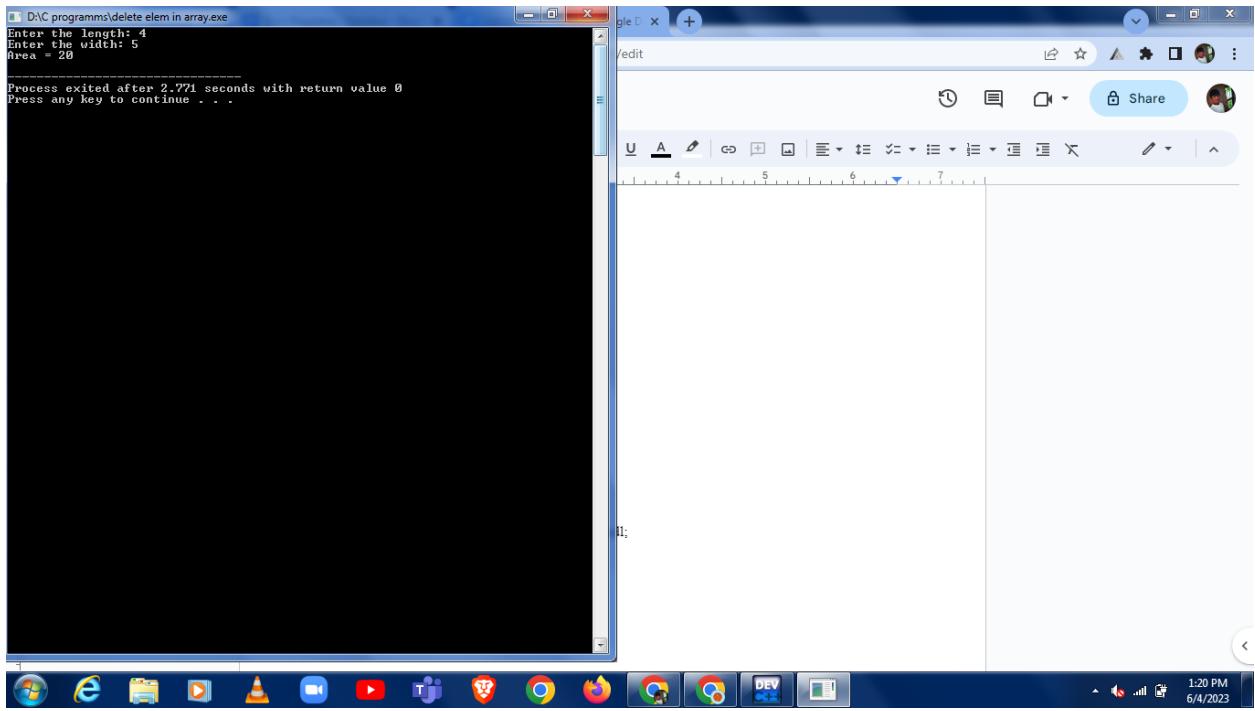
```
{
```

```
private:
```

```
    double length;
```

```
double width;
public:
    Rectangle(double l, double w)
    {
        length = l;
        width = w;
    }
    double getArea()
    {
        return length * width;
    }
};

int main()
{
    double length, width;
    std::cout << "Enter the length: ";
    std::cin >> length;
    std::cout << "Enter the width: ";
    std::cin >> width;
    Rectangle rect(length, width);
    double area = rect.getArea();
    std::cout << "Area = " << area << std::endl;
    return 0;
}
```



46.electricity bill using class

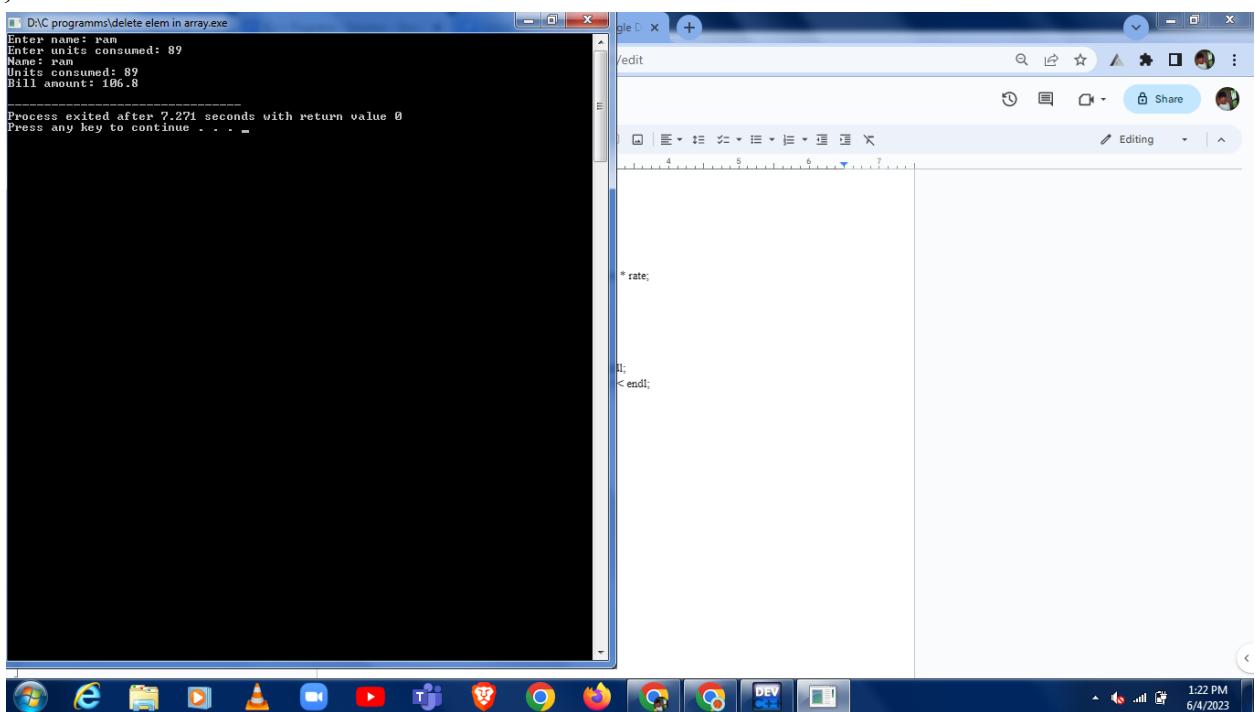
```
#include <iostream>
using namespace std;
class ElectricityBill
{
private:
    string name;
    double units;
public:
    ElectricityBill(string n, double u)
    {
        name = n;
        units = u;
    }
    double calculateBill()
    {
        double rate;
        if (units <= 100)
            rate = 1.20;
        else if (units <= 300)
            rate = 2.00;
        else if (units <= 500)
            rate = 3.00;
```

```

        else
            rate = 5.00;
            double billAmount = units * rate;
            return billAmount;
    }
void displayBill()
{
    cout << "Name: " << name << endl;
    cout << "Units consumed: " << units << endl;
    cout << "Bill amount: " << calculateBill() << endl;
}
};

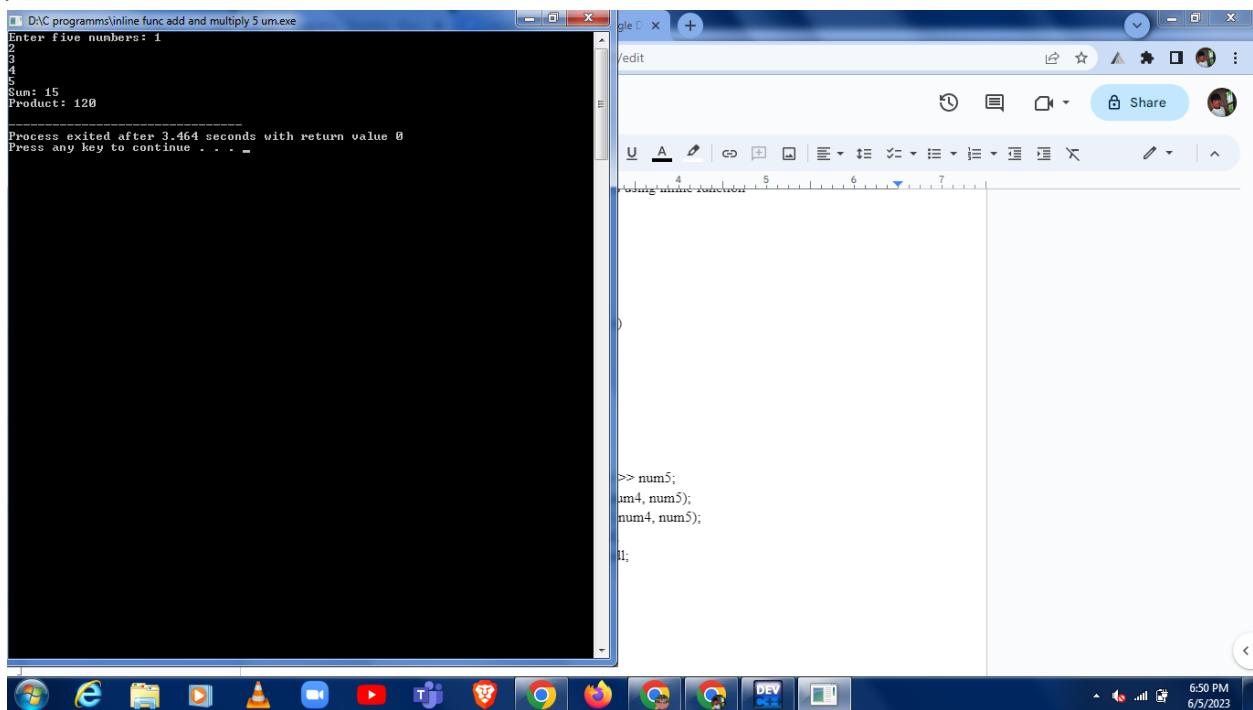
int main() {
    string name;
    double units;
    cout << "Enter name: ";
    cin >> name;
    cout << "Enter units consumed: ";
    cin >> units;
    ElectricityBill bill(name, units);
    bill.displayBill();
    return 0;
}

```

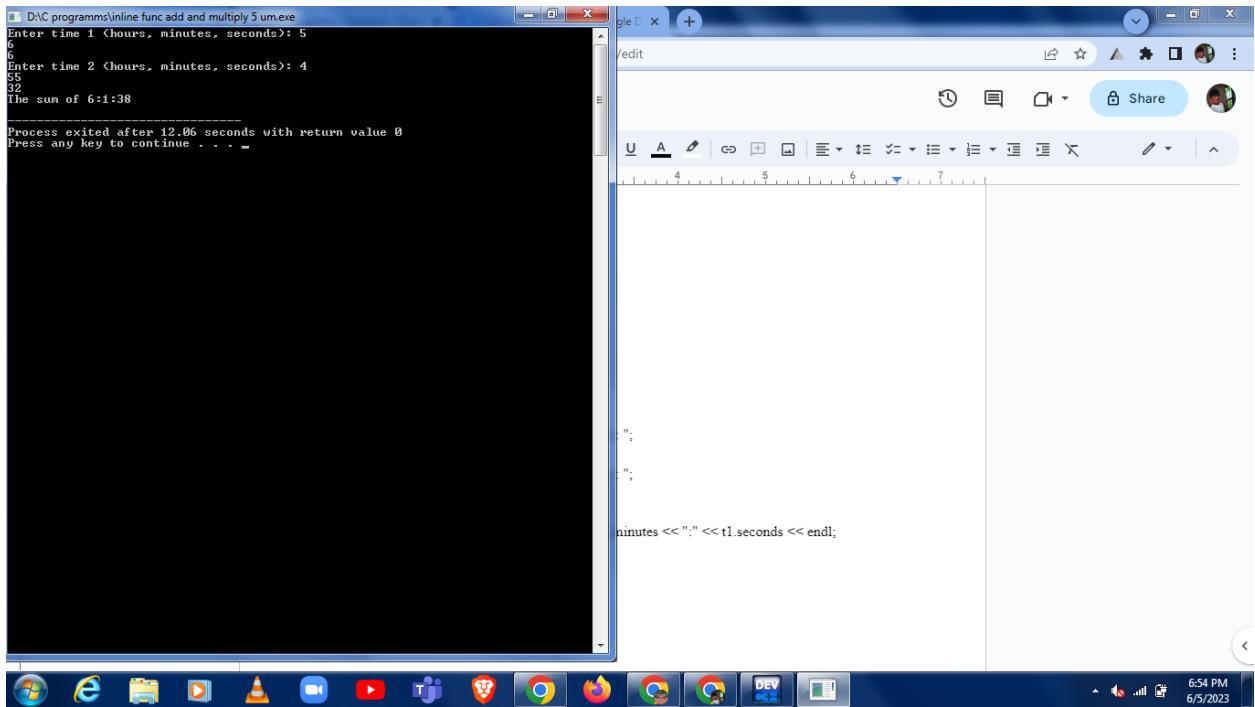


47. program to find sum and product of 5 numbers using inline function

```
#include <iostream>
inline int findSum(int a, int b, int c, int d, int e)
{
    return a + b + c + d + e;
}
inline int findProduct(int a, int b, int c, int d, int e)
{
    return a * b * c * d * e;
}
int main()
{
    int num1, num2, num3, num4, num5;
    std::cout << "Enter five numbers: ";
    std::cin >> num1 >> num2 >> num3 >> num4 >> num5;
    int sum = findSum(num1, num2, num3, num4, num5);
    int product = findProduct(num1, num2, num3, num4, num5);
    std::cout << "Sum: " << sum << std::endl;
    std::cout << "Product: " << product << std::endl;
    return 0;
}
```

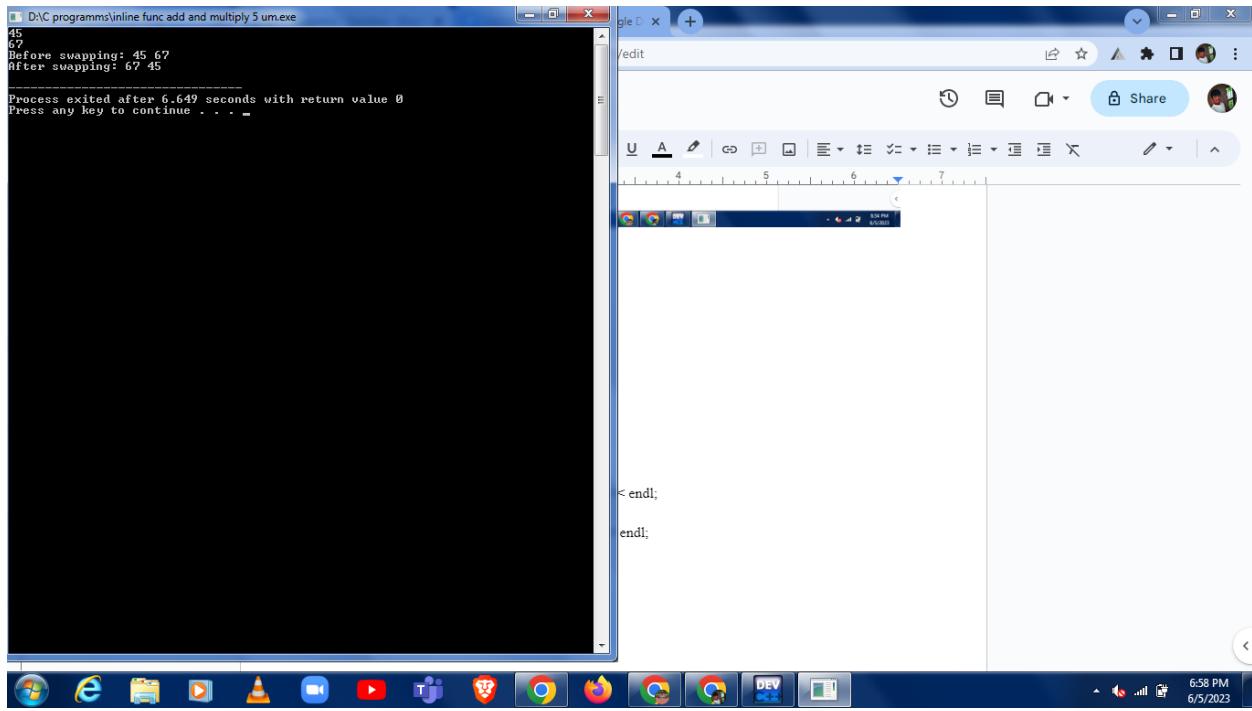


```
48.add time by call by address
#include <iostream>
using namespace std;
struct Time
{
    int hours;
    int minutes;
    int seconds;
};
void addTime(Time *t1, Time *t2)
{
    t1->seconds += t2->seconds;
    if (t1->seconds >= 60)
    {
        t1->seconds -= 60;
        t1->minutes++;
    }
    t1->minutes += t2->minutes;
    if (t1->minutes >= 60)
    {
        t1->minutes -= 60;
        t1->hours++;
    }
}
int main()
{
    Time t1, t2;
    cout << "Enter time 1 (hours, minutes, seconds): ";
    cin >> t1.hours >> t1.minutes >> t1.seconds;
    cout << "Enter time 2 (hours, minutes, seconds): ";
    cin >> t2.hours >> t2.minutes >> t2.seconds;
    addTime(&t1, &t2);
    cout << "The sum = " << t1.hours << ":" << t1.minutes << ":" << t1.seconds << endl;
    return 0;
}
```



49.swap by call by address

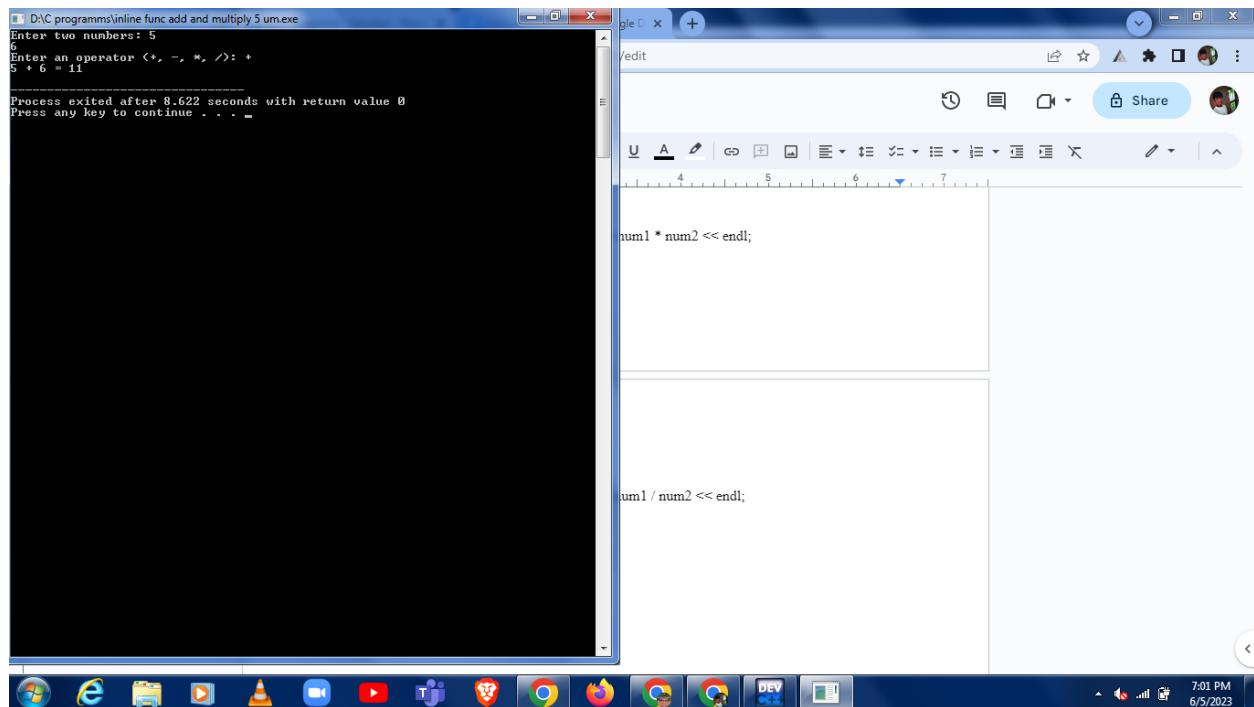
```
#include <iostream>
using namespace std;
void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}
int main()
{
    int a, b;
    cin >> a >> b;
    cout << "Before swapping: " << a << " " << b << endl;
    swap(&a, &b);
    cout << "After swapping: " << a << " " << b << endl;
    return 0;
}
```



50. program for various Mathematical Operations using Switch case

```
#include <iostream>
using namespace std;
int main()
{
    int num1, num2;
    char operation;
    cout << "Enter two numbers: ";
    cin >> num1 >> num2;
    cout << "Enter an operator (+, -, *, /): ";
    cin >> operation;
    switch (operation)
    {
        case '+':
            cout << num1 << " + " << num2 << " = " << num1 + num2 << endl;
            break;
        case '-':
            cout << num1 << " - " << num2 << " = " << num1 - num2 << endl;
            break;
        case '*':
            cout << num1 << " * " << num2 << " = " << num1 * num2 << endl;
            break;
```

```
case '/':  
    cout << num1 << " / " << num2 << " = " << num1 / num2 << endl;  
    break;  
}  
return 0;  
}
```



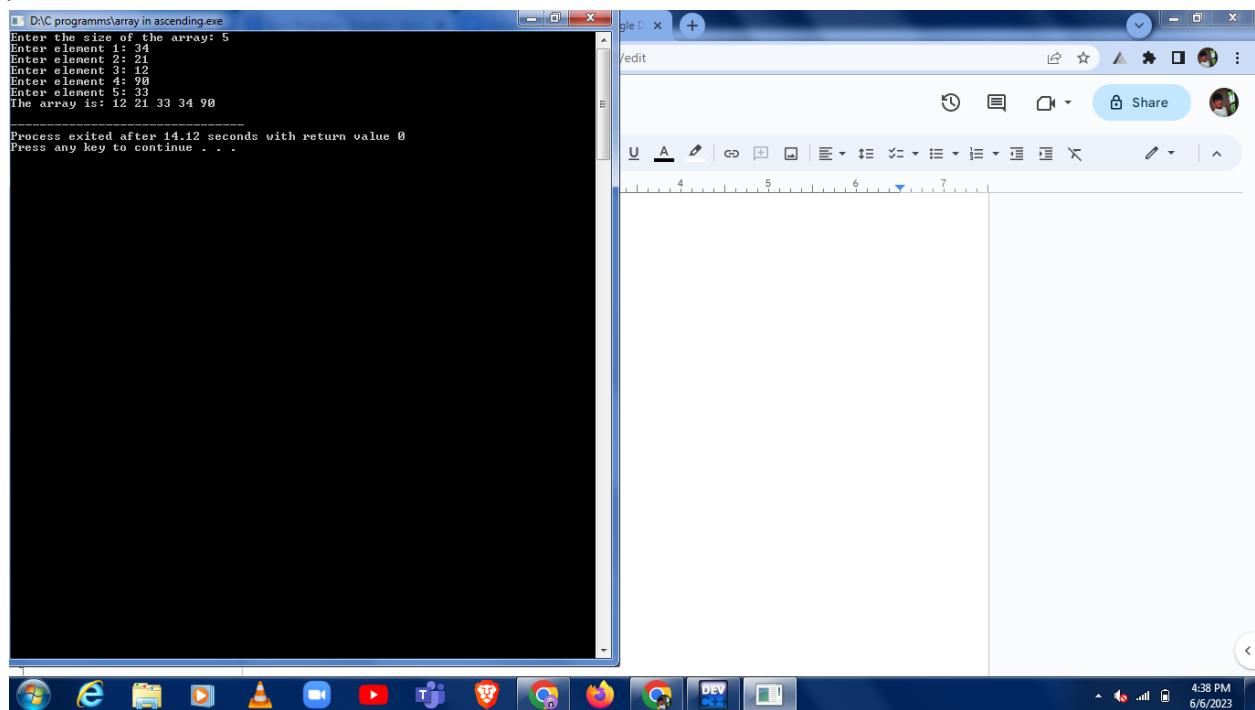
51. Program to Sort an Array Elements in Ascending Order

```
#include <iostream>
using namespace std;
int main()
{
    int size;
    cout << "Enter the size of the array: ";
    cin >> size;
    int array[size];
    for (int i = 0; i < size; i++)
    {
        cout << "Enter element " << i + 1 << ": ";
        cin >> array[i];
    }
    for (int i = 0; i < size - 1; i++)
```

```

{
    for (int j = i + 1; j < size; j++)
    {
        if (array[i] > array[j])
        {
            int temp = array[i];
            array[i] = array[j];
            array[j] = temp;
        }
    }
}
cout << "The array is: ";
for (int i = 0; i < size; i++)
{
    cout << array[i] << " ";
}
cout << endl;
return 0;
}

```



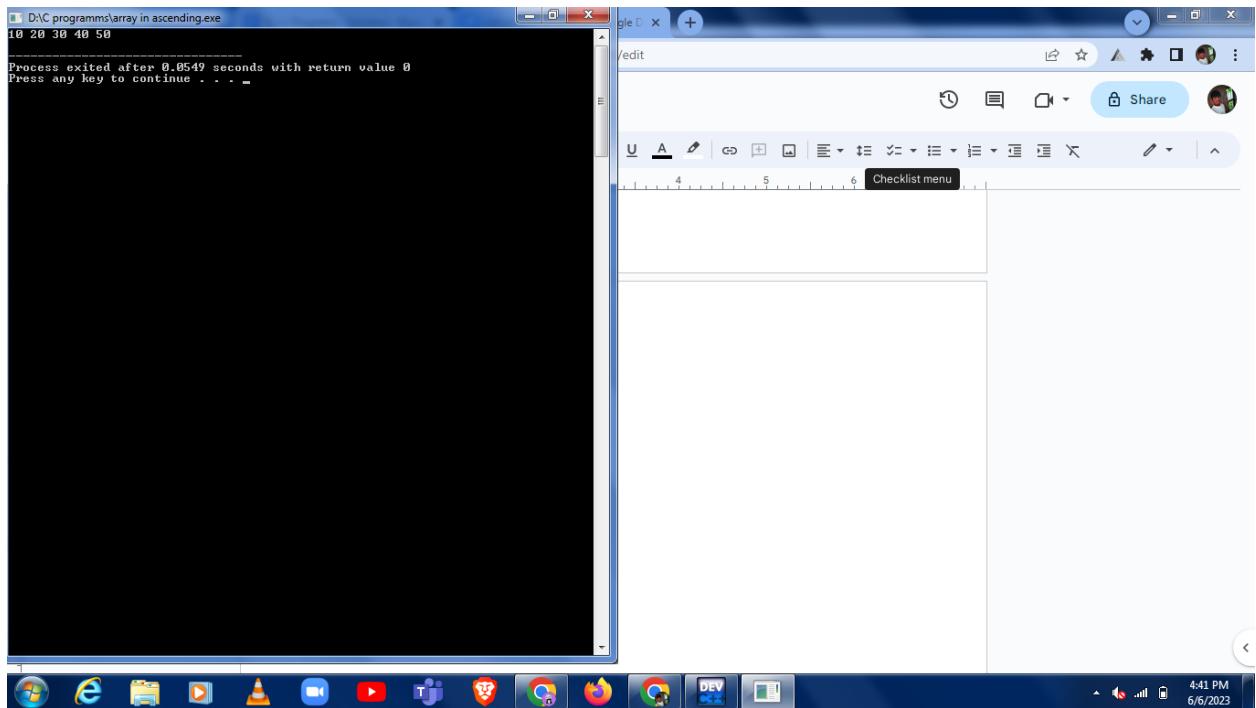
52.access elements in array using ptr

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

```

int main()
{
    int arr[5] = {10, 20, 30, 40, 50};
    int *ptr = arr;
    for (int i = 0; i < 5; i++)
    {
        cout << *ptr << " ";
        ptr++;
    }
    cout << endl;
    return 0;
}

```



53.divisors of a number

```

#include <iostream>
using namespace std;
void printdivisors(int n)
{
    for (int i = 1; i <= n; i++)
    {
        if (n % i == 0)
        {
            cout << i << " ";
        }
    }
}

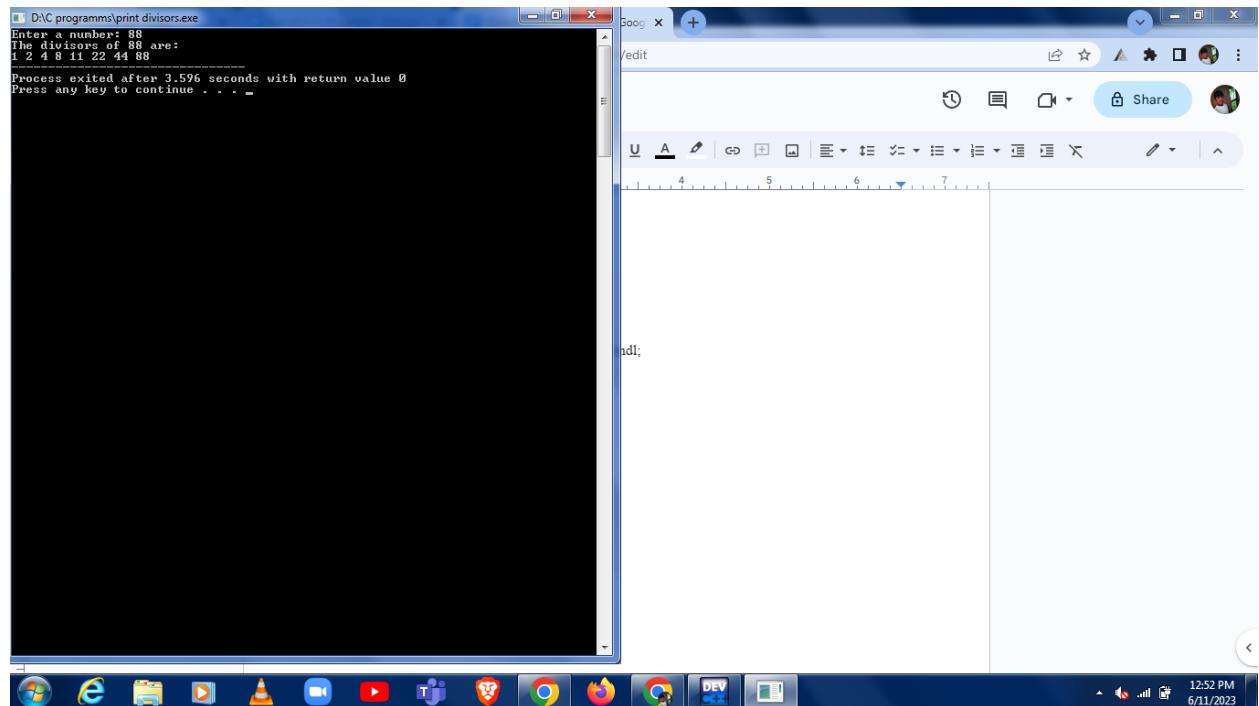
```

```

        }
    }
}

int main()
{
    int n;
    cout << "Enter a number: ";
    cin >> n;
    cout << "The divisors of " << n << " are: " << endl;
    printdivisors(n);
    return 0;
}

```



54.print num from 1 to n

```

#include <iostream>
using namespace std;
class Printnum
{
public:
    void printNumbers(int n)
    {
        for (int i = 1; i <= n; i++)
        {

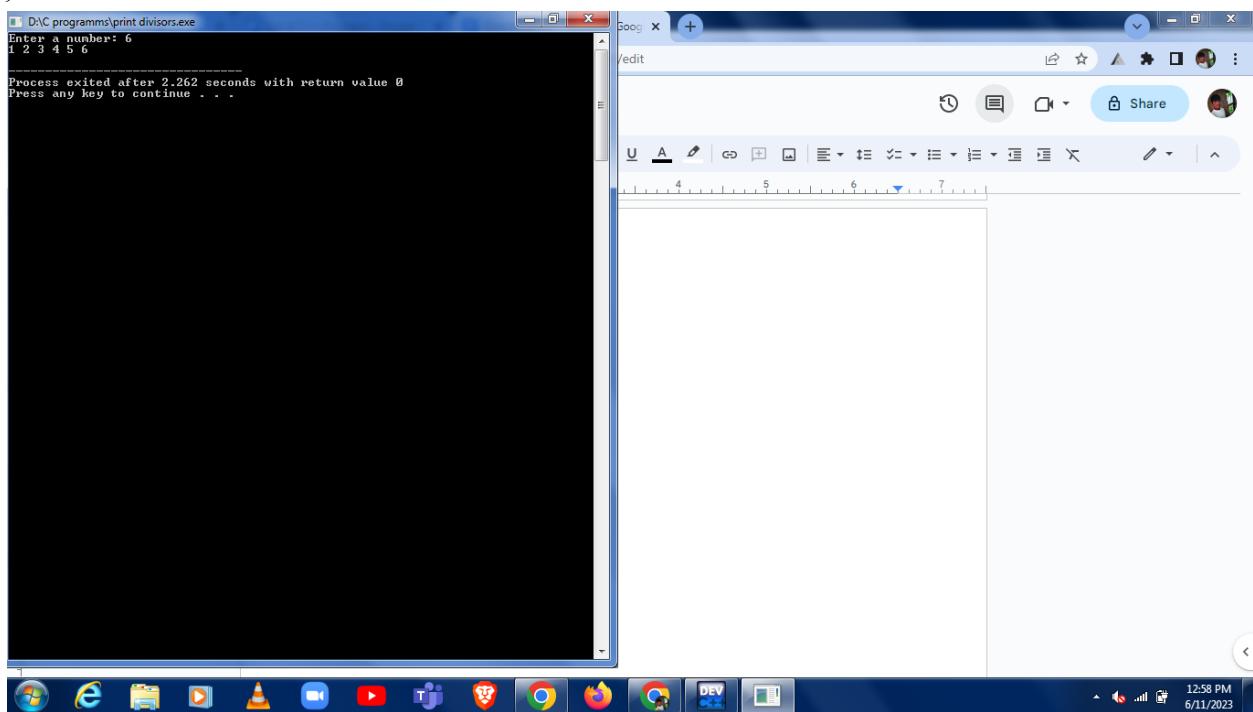
```

```

        cout << i << " ";
    }
    cout << endl;
}
};

int main()
{
    int n;
    cout << "Enter a number: ";
    cin >> n;
    Printnum printer;
    printer.printNumbers(n);
    return 0;
}

```



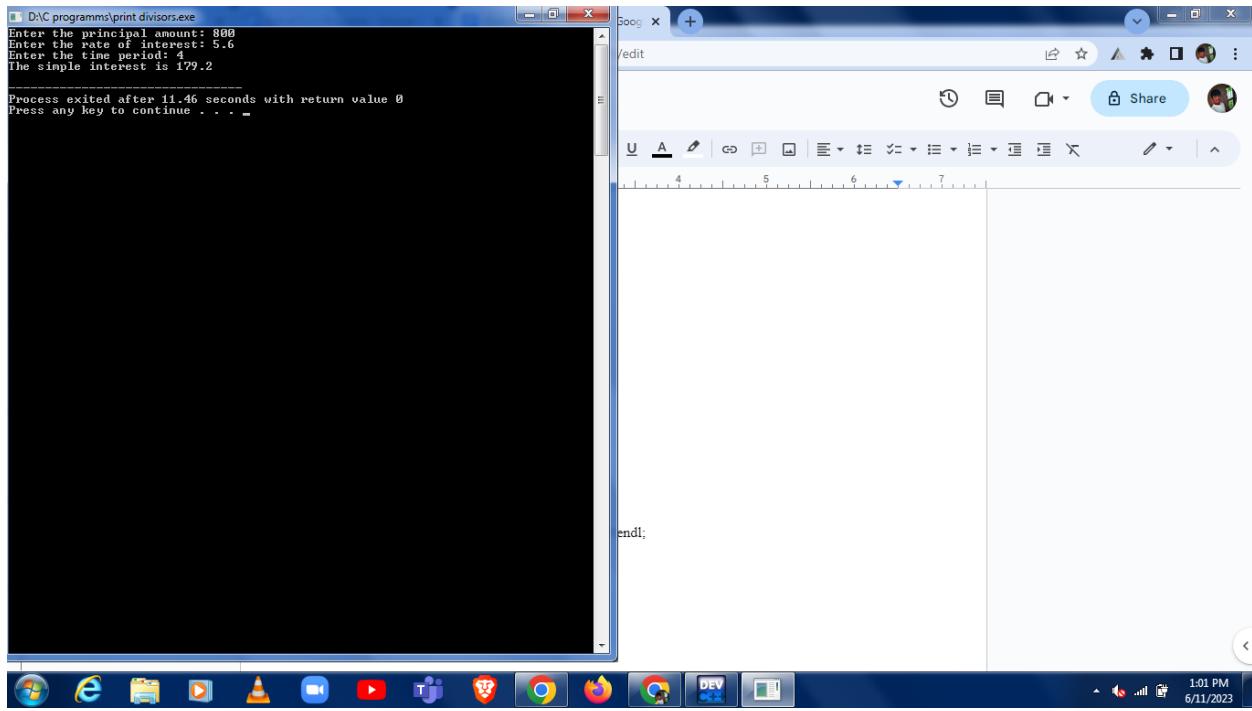
55.cal SI using class

```

#include <iostream>
using namespace std;
class SI
{
public:
    float principal;

```

```
float rate;
float time;
SI(float p, float r, float t)
{
    principal = p;
    rate = r;
    time = t;
}
float calculateInterest()
{
    return (principal * rate * time) / 100;
}
int main()
{
    float principal, rate, time;
    cout << "Enter the principal amount: ";
    cin >> principal;
    cout << "Enter the rate of interest: ";
    cin >> rate;
    cout << "Enter the time period: ";
    cin >> time;
    SI si(principal, rate, time);
    float interest = si.calculateInterest();
    cout << "The simple interest is " << interest << endl;
    return 0;
}
```



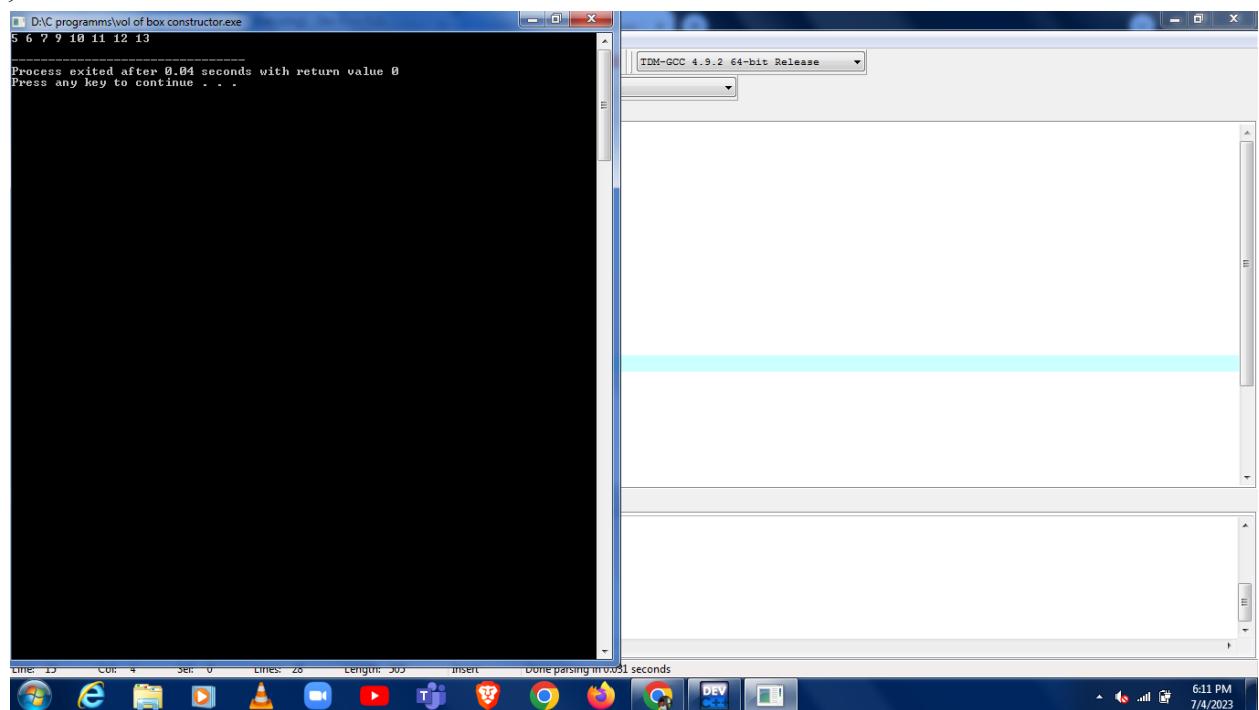
56.insertion sort

```
#include <iostream>
using namespace std;
void insertionSort(int arr[], int n)
{
    for (int i = 1; i < n; i++)
    {
        int key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
    }
}
int main()
{
    int arr[] = {12, 11, 13, 5, 6, 7, 10, 9};
    int n = sizeof(arr) / sizeof(arr[0]);
    insertionSort(arr, n);
```

```

for (int i=0;i<n;i++)
{
    cout<< arr[i] << " ";
}
cout << endl;
return 0;
}

```



57.vol of box constructor

```

#include <iostream>
using namespace std;
class Box
{
public:
    int length;
    int width;
    int height;
    Box(int l, int w, int h)
    {
        length = l;
        width = w;
        height = h;
    }
}

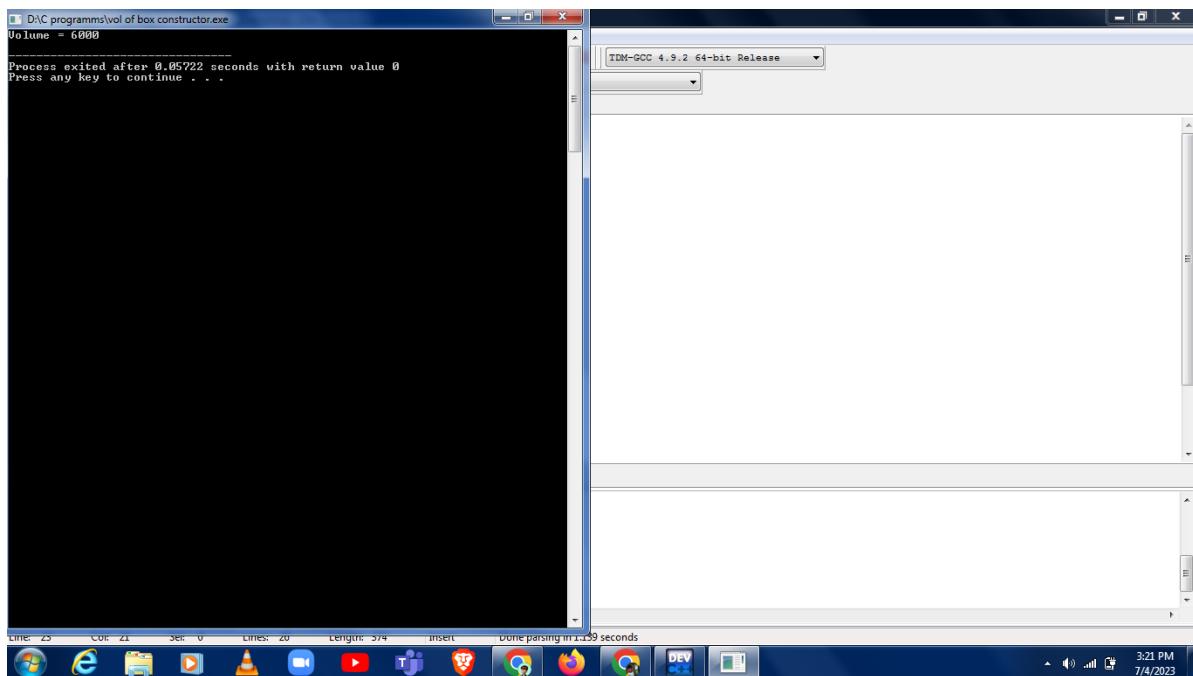
```

```

int volume()
{
    return length * width * height;
}
};

int main()
{
    Box box(10, 20, 30);
    cout << "Volume = " << box.volume() << endl;
    return 0;
}

```



58.area of rectangle using inheritance

```

#include <iostream>
using namespace std;
class Shape
{
protected:
    int width;
    int height;
public:
    Shape() {}
    Shape(int width, int height)
    {

```

```

        this->width = width;
        this->height = height;
    }
    virtual int getArea() = 0;
};

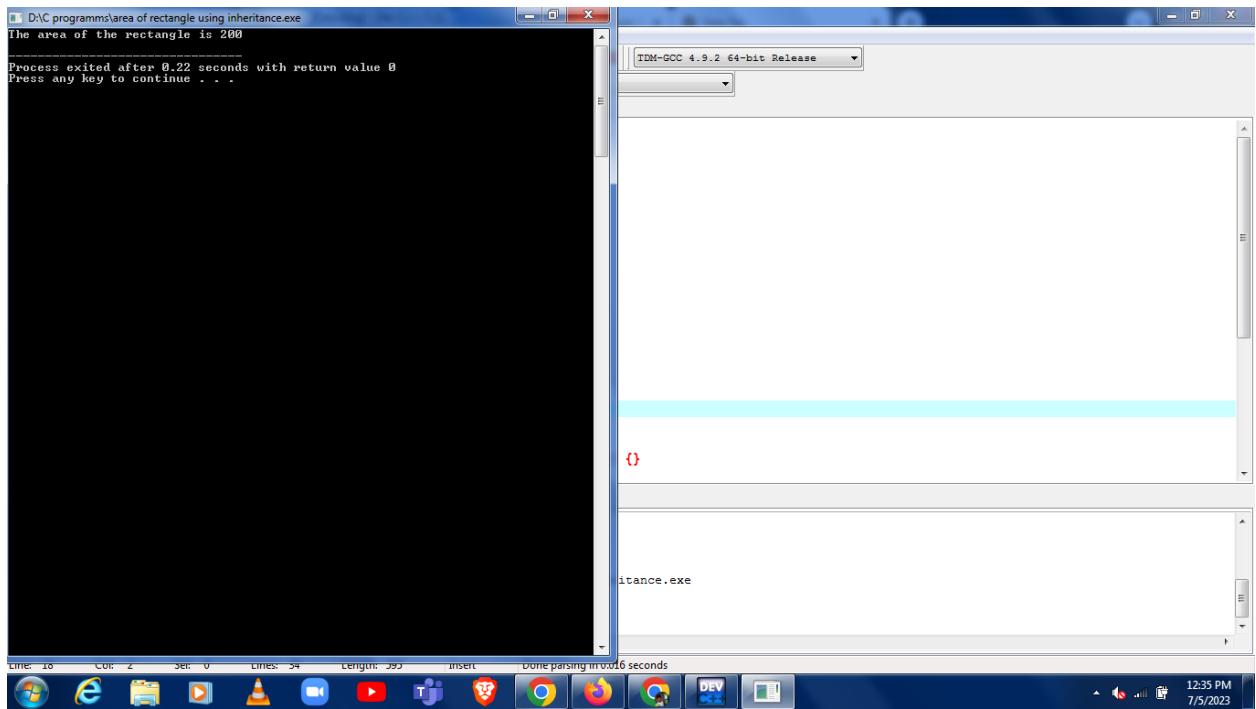
class Rectangle : public Shape
{
public:
    Rectangle() {}

    Rectangle(int width, int height) : Shape(width, height) {}

    int getArea()
    {
        return width * height;
    }
};

int main()
{
    Rectangle value(10, 20);
    int area = value.getArea();
    cout << "The area of the rectangle is " << area << endl;
    return 0;
}

```

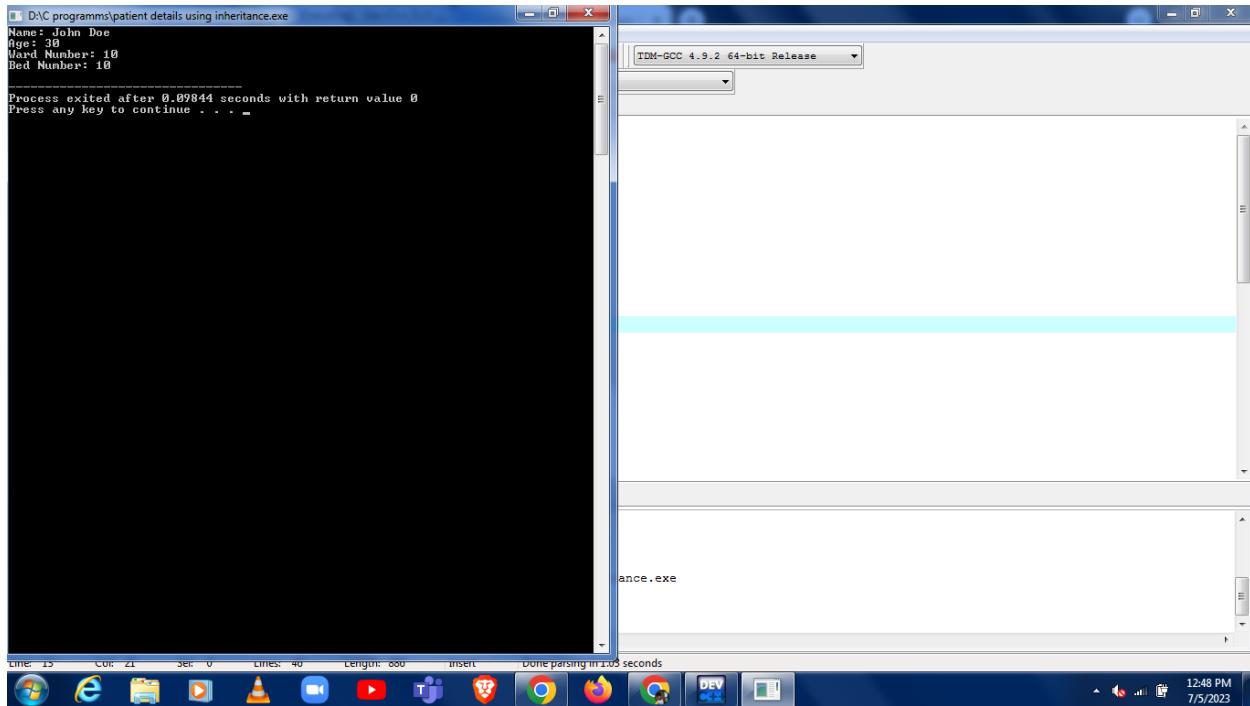


```
59.patient details using inheritance
#include <iostream>
using namespace std;
class Patient
{
protected:
    string name;
    int age;
public:
    Patient() {}
    Patient(string name, int age)
    {
        this->name = name;
        this->age = age;
    }
    void printDetails()
    {
        cout << "Name: " << name << endl;
        cout << "Age: " << age << endl;
    }
};
class NewPatient : public Patient
{
private:
    int wardNumber;
    int bedNumber;
public:
    NewPatient() {}
    NewPatient(string name, int age, int wardNumber, int bedNumber) : Patient(name, age)
    {
        this->wardNumber = wardNumber;
        this->bedNumber = bedNumber;
    }
    void printDetails()
    {
        Patient::printDetails();
        cout << "Ward Number: " << wardNumber << endl;
        cout << "Bed Number: " << bedNumber << endl;
    }
};
```

```

int main()
{
    NewPatient info("John", 30, 10, 10);
    info.printDetails();
    return 0;
}

```



60.area and volume using inheritance

```

#include <iostream>
using namespace std;
class Shape
{
protected:
    int width;
    int height;
public:
    Shape() {}
    Shape(int width, int height)
    {
        this->width = width;
        this->height = height;
    }
    virtual int getArea() = 0;

```

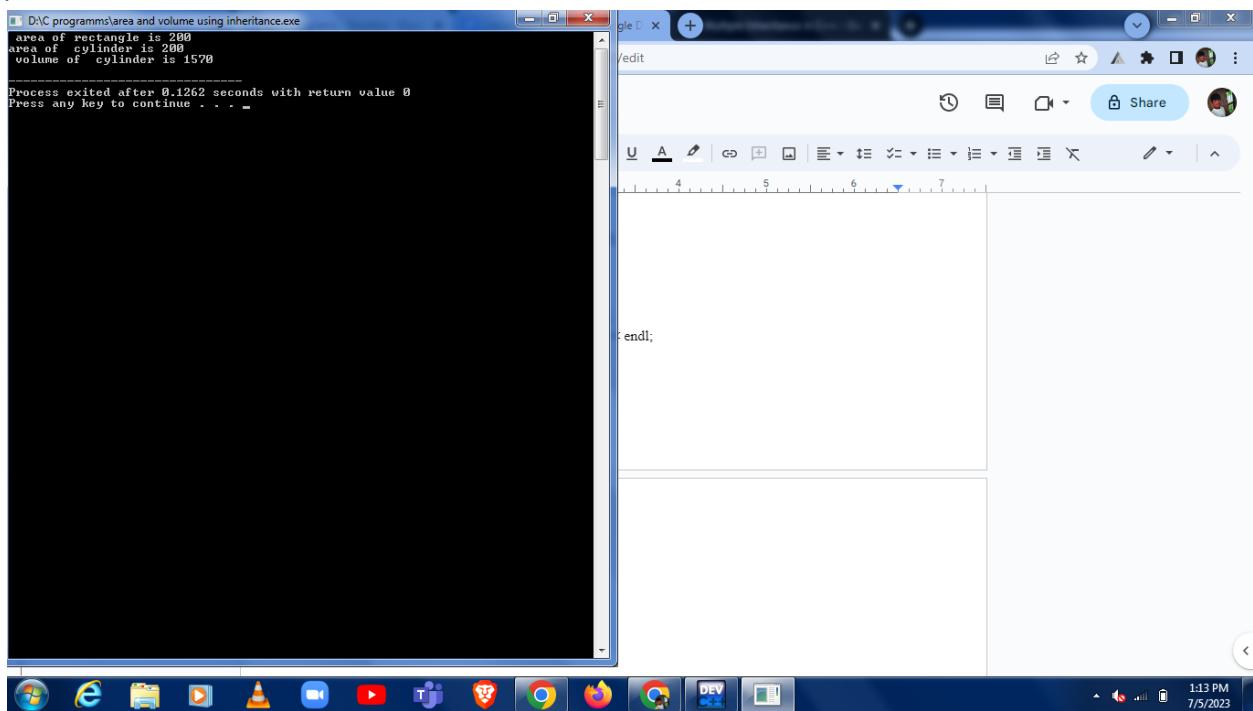
```
};

class Rectangle : public Shape
{
public:
    Rectangle() {}
    Rectangle(int width, int height) : Shape(width, height) {}
    int getArea()
    {
        return width * height;
    }
};

class Cylinder : public Shape
{
protected:
    int radius;
public:
    Cylinder() {}
    Cylinder(int width, int height, int radius) : Shape(width, height)
    {
        this->radius = radius;
    }
    int getArea()
    {
        return 2 * 3.14 * radius * radius + 2 * 3.14 * radius * height;
    }
    int getVolume()
    {
        return 3.14 * radius * radius * height;
    }
};

int main() {
    Rectangle info(10, 20);
    int area = info.getArea();
    cout << " area of rectangle is " << area << endl;
    Cylinder value(10, 20, 5);
    int Area = value.getArea();
    int volume = value.getVolume();
    cout << "area of cylinder is " << area << endl;
    cout << " volume of cylinder is " << volume << endl;
    return 0;
}
```

```
}
```



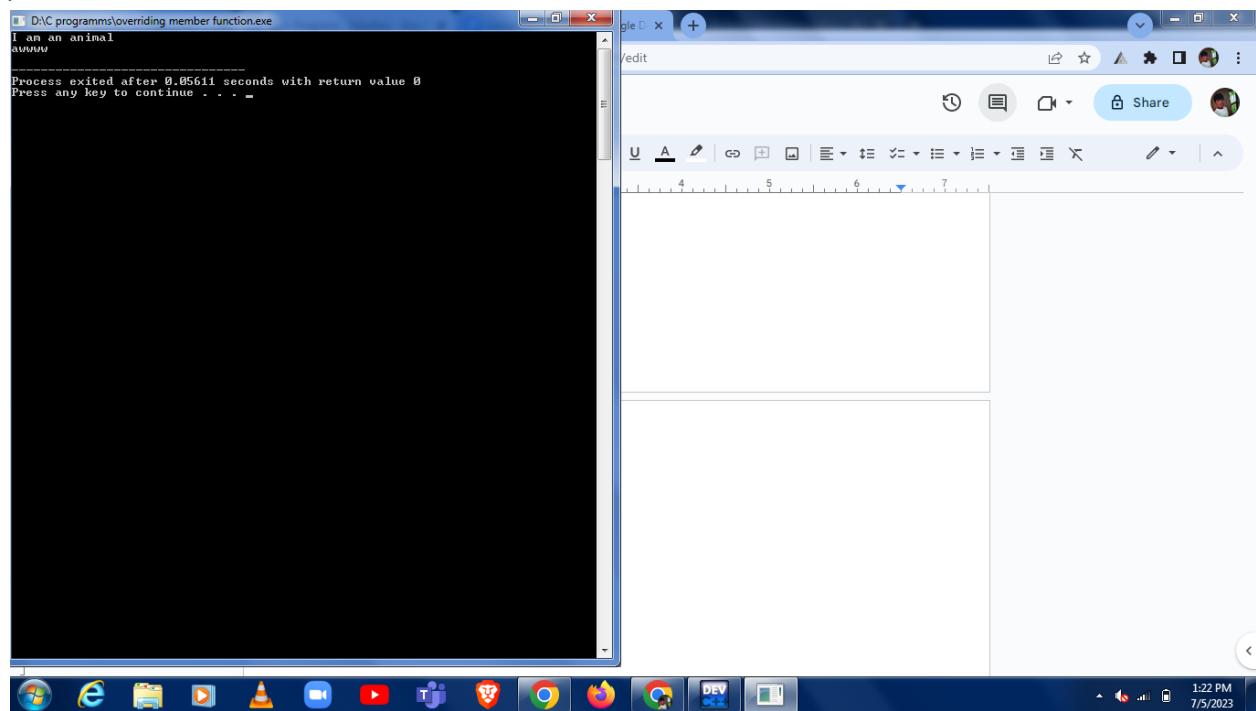
61.overriding member function

```
#include <iostream>
using namespace std;
class Animal
{
public:
    void speak()
    {
        cout << "I am an animal" << endl;
    }
};
class Dog : public Animal
{
public:
    void speak() {
        cout << "awwww" << endl;
    }
};
int main()
{
    Animal *animal = new Animal();
```

```

animal->speak();
Dog *dog = new Dog();
dog->speak();
return 0;
}

```



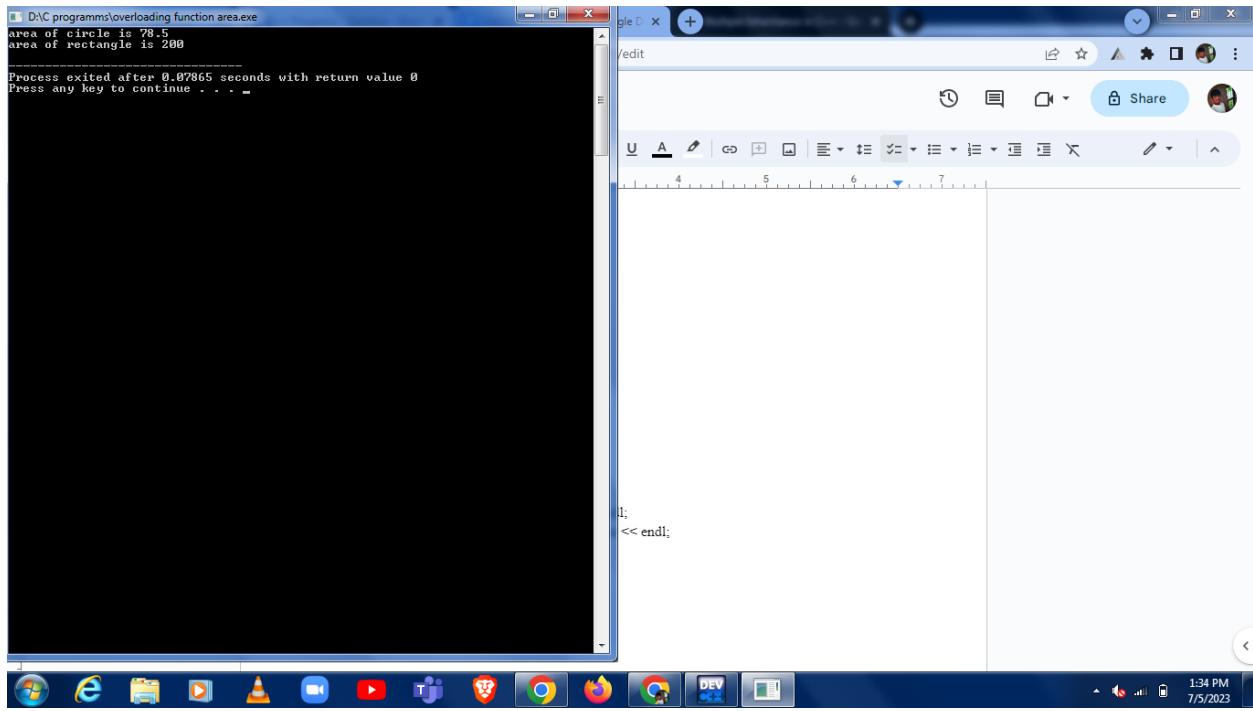
62. overloading in function to find area

```

#include <iostream>
using namespace std;
class Shape {
public:
    virtual double getArea() = 0;
};
class Circle : public Shape
{
private:
    double radius;
public:
    Circle(double radius)
    {
        this->radius = radius;
    }
    double getArea()

```

```
{  
    return 3.14 * radius * radius;  
}  
};  
class Rectangle : public Shape  
{  
private:  
    double width;  
    double height;  
public:  
    Rectangle(double width, double height)  
    {  
        this->width = width;  
        this->height = height;  
    }  
    double getArea()  
    {  
        return width * height;  
    }  
};  
int main()  
{  
    Circle circle(5);  
    Rectangle rectangle(10, 20);  
    double circleArea = circle.getArea();  
    double rectangleArea = rectangle.getArea();  
    cout << "area of circle is " << circleArea << endl;  
    cout << "area of rectangle is " << rectangleArea << endl;  
    return 0;  
}
```



63. volume cube sphere cylinder overloading function

```
#include <iostream>
using namespace std;
class Shape
{
public:
    virtual double getVolume() = 0;
};
class Cube : public Shape
{
private:
    double side;
public:
    Cube(double side)
    {
        this->side = side;
    }
    double getVolume()
    {
        return side * side * side;
    }
};
```

```
class Cylinder : public Shape
{
private:
    double radius;
    double height;
public:
    Cylinder(double radius, double height)
    {
        this->radius = radius;
        this->height = height;
    }
    double getVolume()
    {
        return 3.14 * radius * radius * height;
    }
};

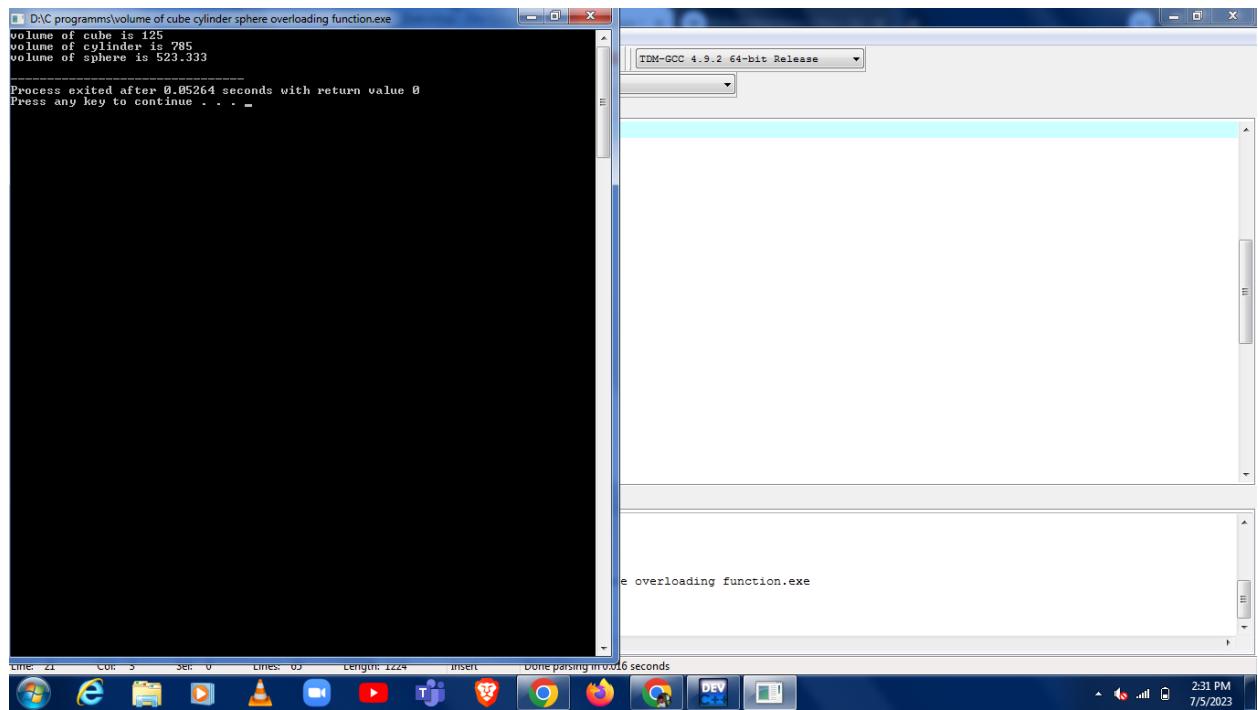
class Sphere : public Shape
{
private:
    double radius;
public:
    Sphere(double radius)
    {
        this->radius = radius;
    }
    double getVolume()
    {
        return (4.0 / 3.0) * 3.14 * radius * radius * radius;
    }
};

int main()
{
    Cube cube(5);
    Cylinder cylinder(5, 10);
    Sphere sphere(5);
    double cubeVolume = cube.getVolume();
    double cylinderVolume = cylinder.getVolume();
    double sphereVolume = sphere.getVolume();
    cout << "volume of cube is " << cubeVolume << endl;
    cout << "volume of cylinder is " << cylinderVolume << endl;
```

```

cout << "volume of sphere is " << sphereVolume << endl;
return 0;
}

```



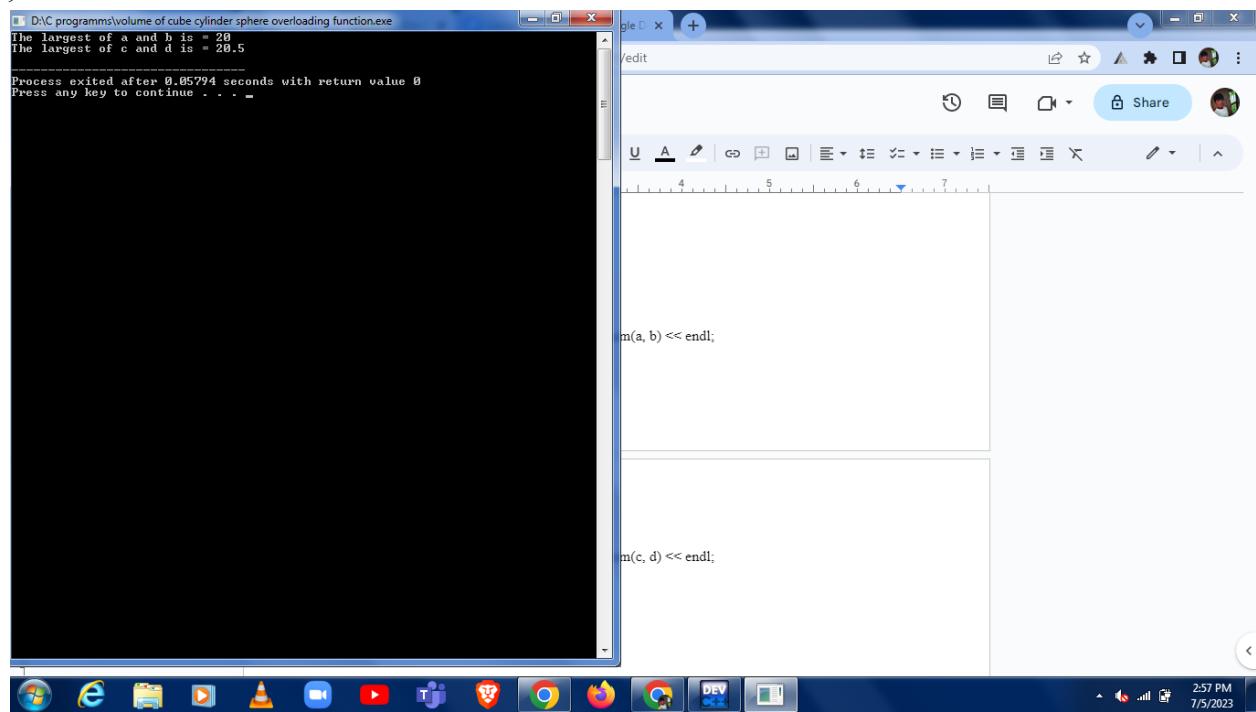
64. largest num of 2 using template

```

#include <iostream>
using namespace std;
template <class T>
T largenum(T x, T y)
{
    if (x > y)
    {
        return x;
    } else
    {
        return y;
    }
}
int main()
{
    int a = 10, b = 20;
    float c = 10.5, d = 20.5;
    cout << "The largest of a and b is = " << largenum(a, b) << endl;
}

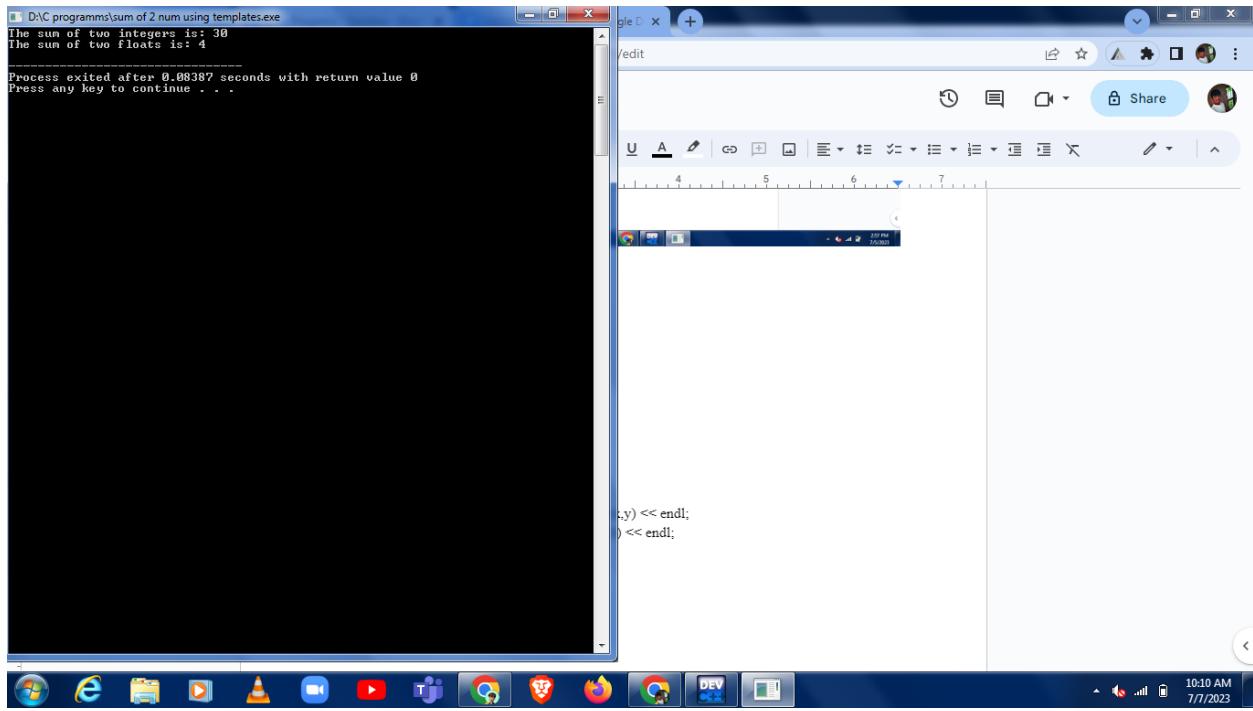
```

```
cout << "The largest of c and d is = " << largenum(c, d) << endl;
return 0;
}
```



65.sum of 2 num using template

```
#include <iostream>
using namespace std;
template <typename T>
T sum(T a, T b)
{
    return a + b;
}
int main()
{
    int x = 10, y = 20;
    float a = 1.5, b = 2.5;
    cout << "The sum of two integers is: " << sum(x,y) << endl;
    cout << "The sum of two floats is: " << sum(a,b) << endl;
    return 0;
}
```



66.student details different stream inheritance

```
#include <iostream>
class Student
{
protected:
    int roll_number;
    std::string name;
public:
    Student(int roll_number, std::string name)
    {
        this->roll_number = roll_number;
        this->name = name;
    }
    void print_details()
    {
        std::cout << "Roll number: " << roll_number << std::endl;
        std::cout << "Name: " << name << std::endl;
    }
};
class CSEStudent : public Student
{
protected:
```

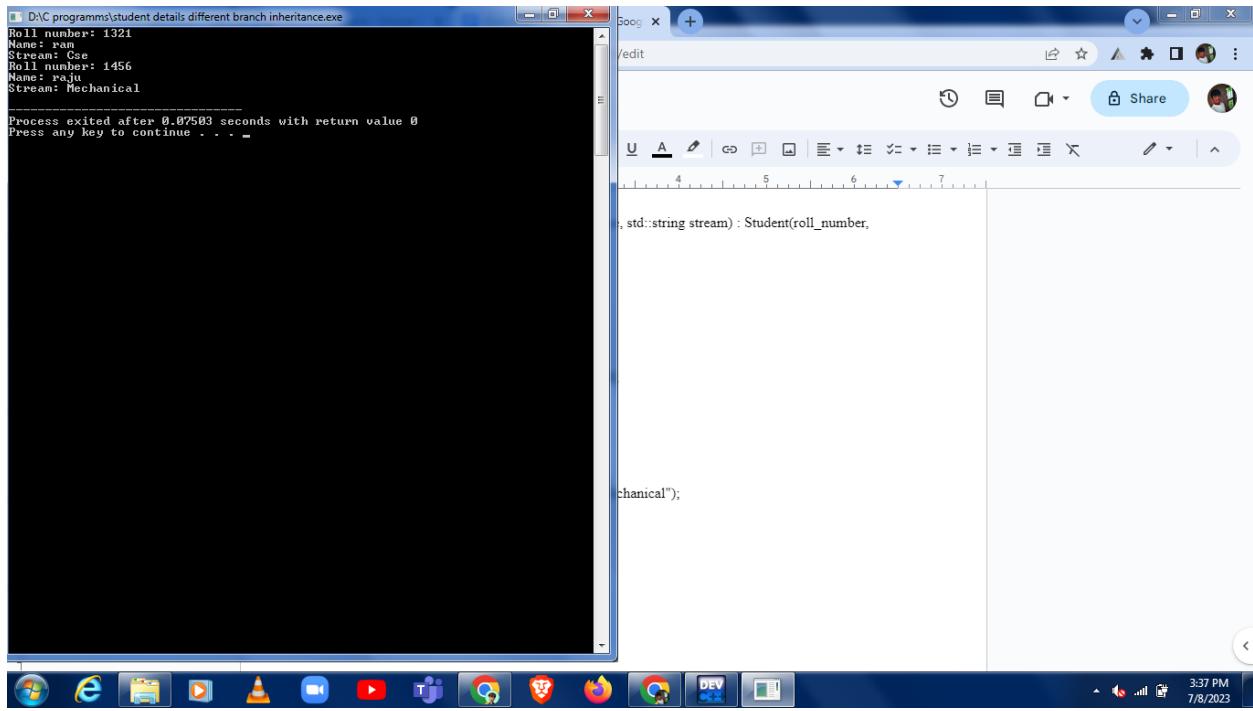
```

std::string stream;
public:
    CSEStudent(int roll_number, std::string name, std::string stream) : Student(roll_number, name)
    {
        this->stream = stream;
    }
    void print_details()
    {
        Student::print_details();
        std::cout << "Stream: " << stream << std::endl;
    }
};

class MECHStudent : public Student
{
protected:
    std::string stream;
public:
    MECHStudent(int roll_number, std::string name, std::string stream) : Student(roll_number,
name)
    {
        this->stream = stream;
    }
    void print_details()
    {
        Student::print_details();
        std::cout << "Stream: " << stream << std::endl;
    }
};

int main()
{
    CSEStudent cse_student(1321, "ram", "Cse");
    MECHStudent mech_student(1456, "raju", "Mechanical");
    cse_student.print_details();
    mech_student.print_details();
    return 0;
}

```



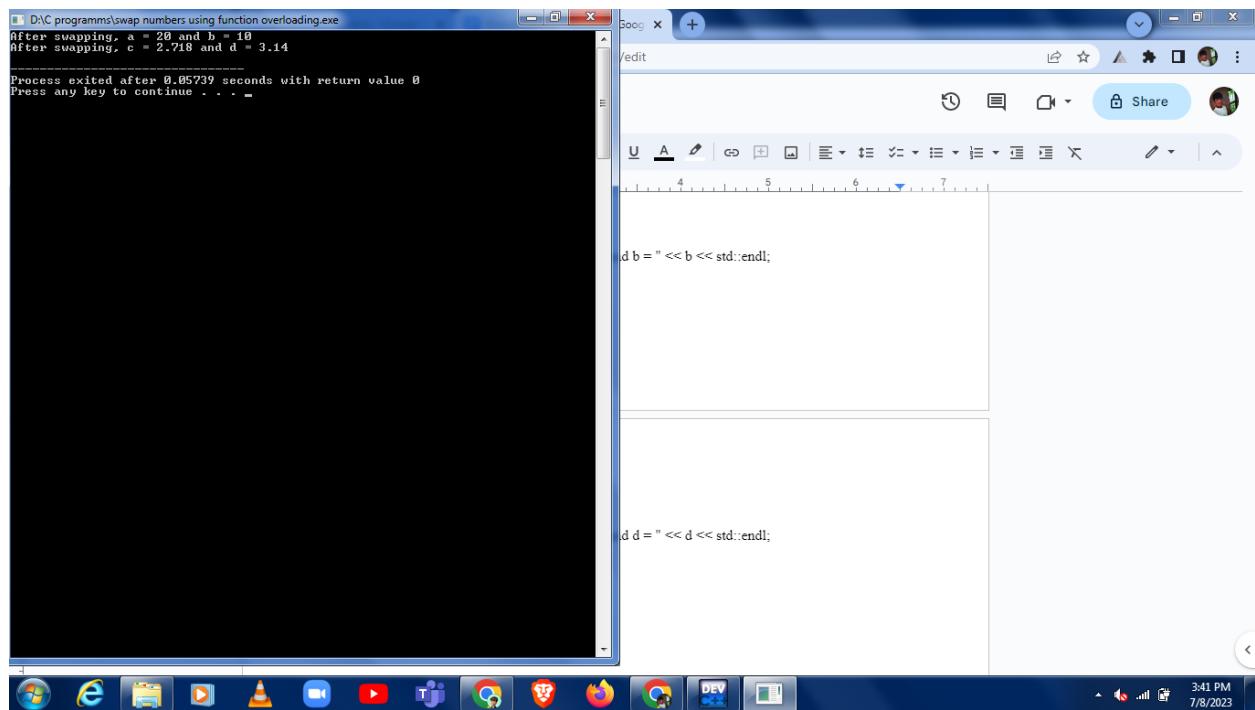
67.swap numbers using overloading

```
#include <iostream>
void swap(int& a, int& b)
{
    int temp = a;
    a = b;
    b = temp;
}
void swap(double& a, double& b)
{
    double temp = a;
    a = b;
    b = temp;
}
int main()
{
    int a = 10;
    int b = 20;
    swap(a, b);
    std::cout << "After swapping, a = " << a << " and b = " << b << std::endl;
    double c = 3.14;
    double d = 2.718;
```

```

    swap(c, d);
    std::cout << "After swapping, c = " << c << " and d = " << d << std::endl;
    return 0;
}

```



68.calculator using template

```

#include <iostream>
using namespace std;
template <class T>
class Calculator
{
private:
    T num1, num2;
public:
    Calculator(T n1, T n2)
    {
        num1 = n1;
        num2 = n2;
    }
    T add()
    {
        return num1 + num2;
    }
}

```

```
T subtract()
{
    return num1 - num2;
}
T multiply()
{
    return num1 * num2;
}
T divide()
{
    return num1 / num2;
}
void displayResult()
{
    cout << "The result is: " << endl;
    cout << num1 << " " << "+" << " " << num2 << " = " << add() << endl;
    cout << num1 << " " << "-" << " " << num2 << " = " << subtract() << endl;
    cout << num1 << " " << "*" << " " << num2 << " = " << multiply() << endl;
    cout << num1 << " " << "/" << " " << num2 << " = " << divide() << endl;
}
};

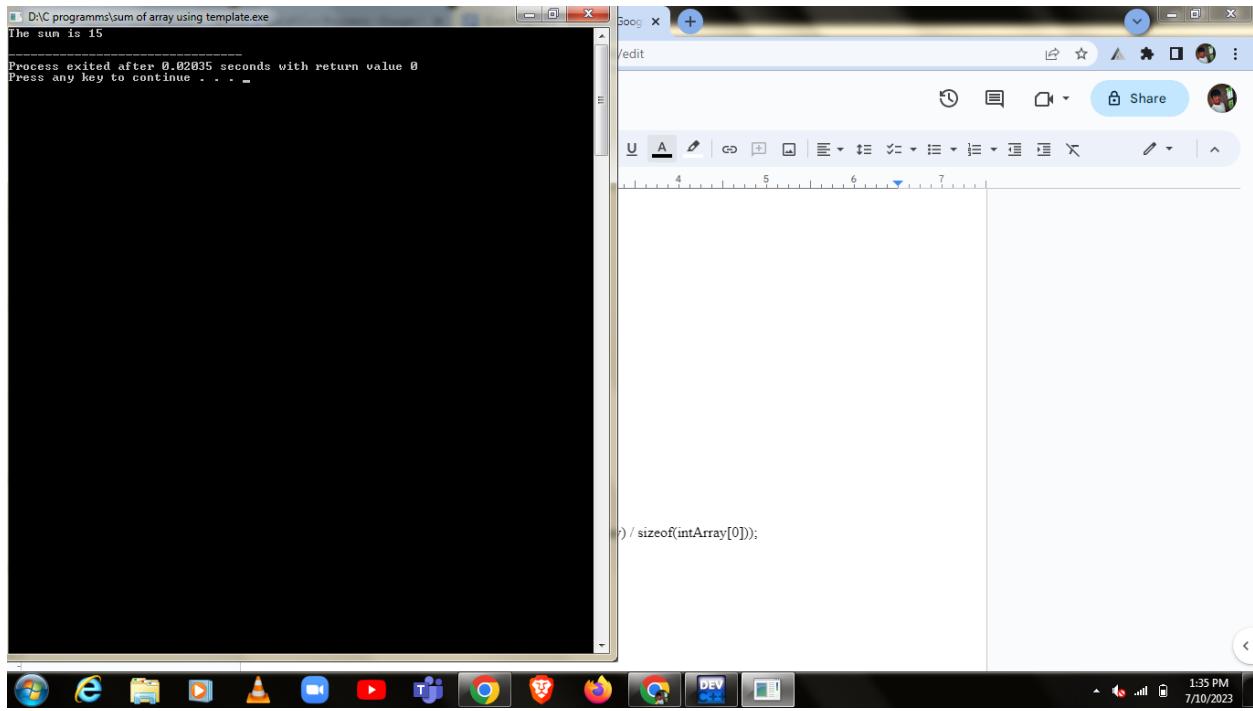
int main()
{
    Calculator<int> intCalc(2,7);
    intCalc.displayResult();
    Calculator<float> floatCalc(2.9,3.2);
    floatCalc.displayResult();
    return 0;
}
```

The screenshot shows a Windows desktop environment. On the left, a terminal window titled 'D:\C programs\calculator using template.exe' displays the output of a calculator program. It shows integer operations (addition, subtraction, multiplication, division) and floating-point operations (addition, subtraction, multiplication, division) between the values 2.9 and 3.2. On the right, a code editor window titled 'TDM-GCC 4.9.2 64-bit Release' shows C++ code. The code defines a template class T with a static member function sumArray that calculates the sum of an array of type T. The main function creates an array of integers, calls sumArray, and prints the result. A specific section of the code in the editor is highlighted in red, showing four overloaded operators: add(), subtract(), multiply(), and divide().

```
The result is:  
2 + 2 = 4  
2 - 2 = 0  
2 * 2 = 4  
2 / 2 = 1  
The result is:  
2.9 + 3.2 = 6.1  
2.9 - 3.2 = -0.3  
2.9 * 3.2 = 9.68  
2.9 / 3.2 = 0.90625  
  
Process exited after 0.163 seconds with return value 0  
Press any key to continue . . .  
  
add() << endl;  
subtract() << endl;  
multiply() << endl;  
divide() << endl;
```

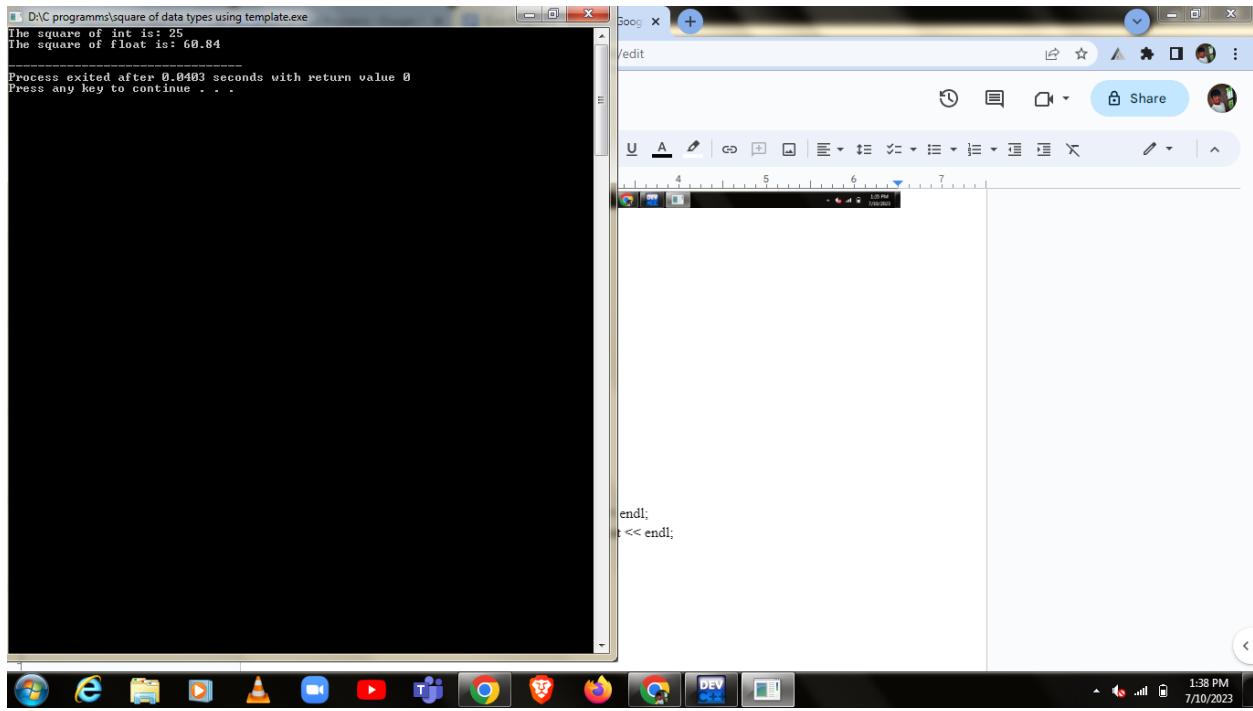
69.sum of array usig template

```
#include <iostream>  
using namespace std;  
template <class T>  
T sumArray(T* array, int size)  
{  
    T sum = 0;  
    for (int i = 0; i < size; i++)  
    {  
        sum += array[i];  
    }  
    return sum;  
}  
int main()  
{  
    int intArray[] = {1,2,3,4,5};  
    int intSum = sumArray(intArray, sizeof(intArray) / sizeof(intArray[0]));  
    cout << "The sum is " << intSum << endl;  
    return 0;  
}
```



70.square of data types using template

```
#include <iostream>
using namespace std;
template <class T>
T square(T num)
{
    return num * num;
}
int main()
{
    int intNum = 5;
    float floatNum = 7.8;
    int squareInt = square<int>(intNum);
    float squareFloat = square<float>(floatNum);
    cout << "The square of int is: " << squareInt << endl;
    cout << "The square of float is: " << squareFloat << endl;
    return 0;
}
```



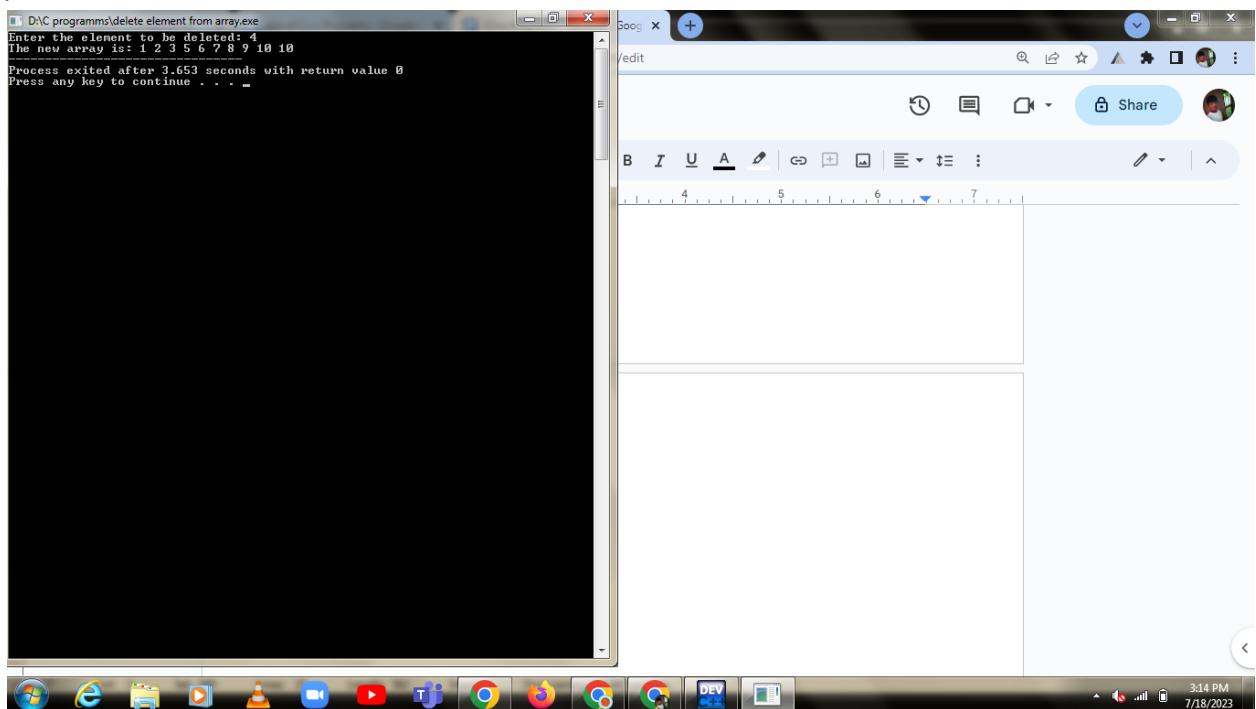
71.delete elem from array

```
#include <iostream>
using namespace std;
int main()
{
    int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
    int elementToDelete;
    cout << "Enter the element to be deleted: ";
    cin >> elementToDelete;
    int index = -1;
    for (int i = 0; i < 10; i++) {
        if (arr[i] == elementToDelete) {
            index = i;
            break;
        }
    }
    if (index != -1) {
        for (int i = index; i < 9; i++) {
            arr[i] = arr[i + 1];
        }
    }
    cout << "The new array is: ";
```

```

for (int i = 0; i < 10; i++) {
    cout << arr[i] << " ";
}
return 0;
}

```

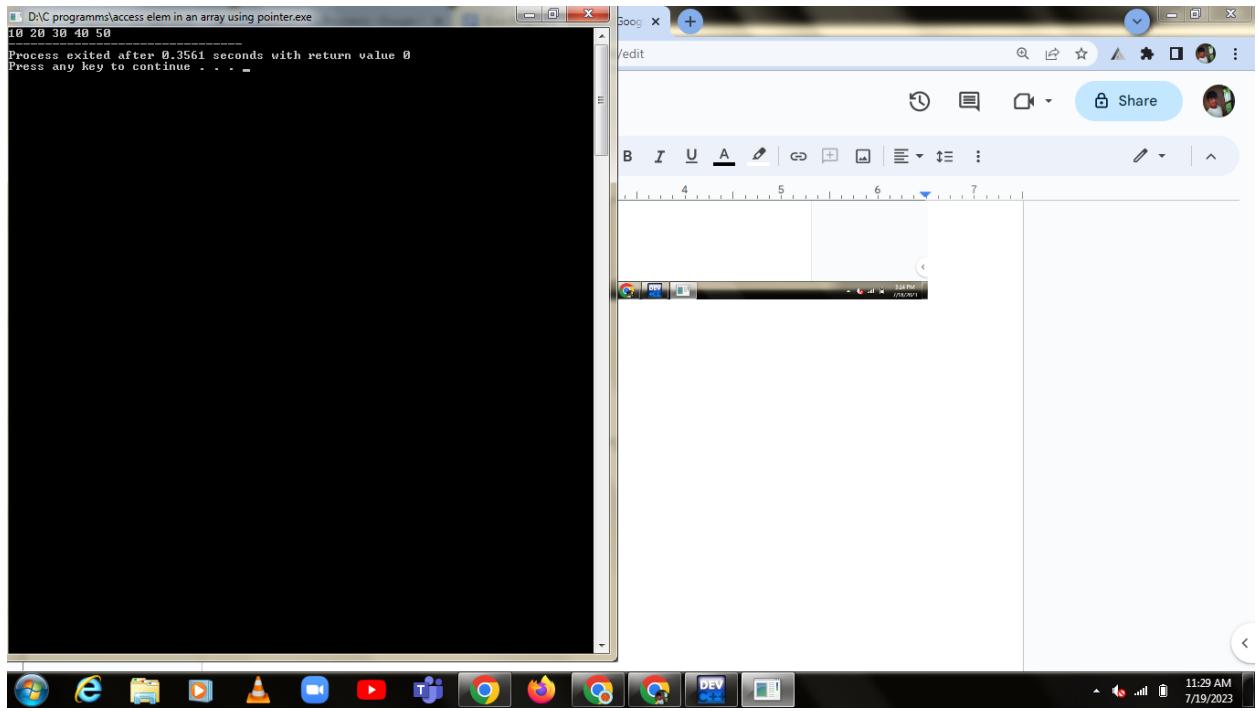


72.access elem in an array using pointer

```

#include <iostream>
using namespace std;
int main() {
    int arr[] = {10, 20, 30, 40, 50};
    int *ptr = arr;
    for (int i = 0; i < sizeof(arr) / sizeof(arr[0]); i++)
    {
        cout << *ptr++ << " ";
    }
    return 0;
}

```



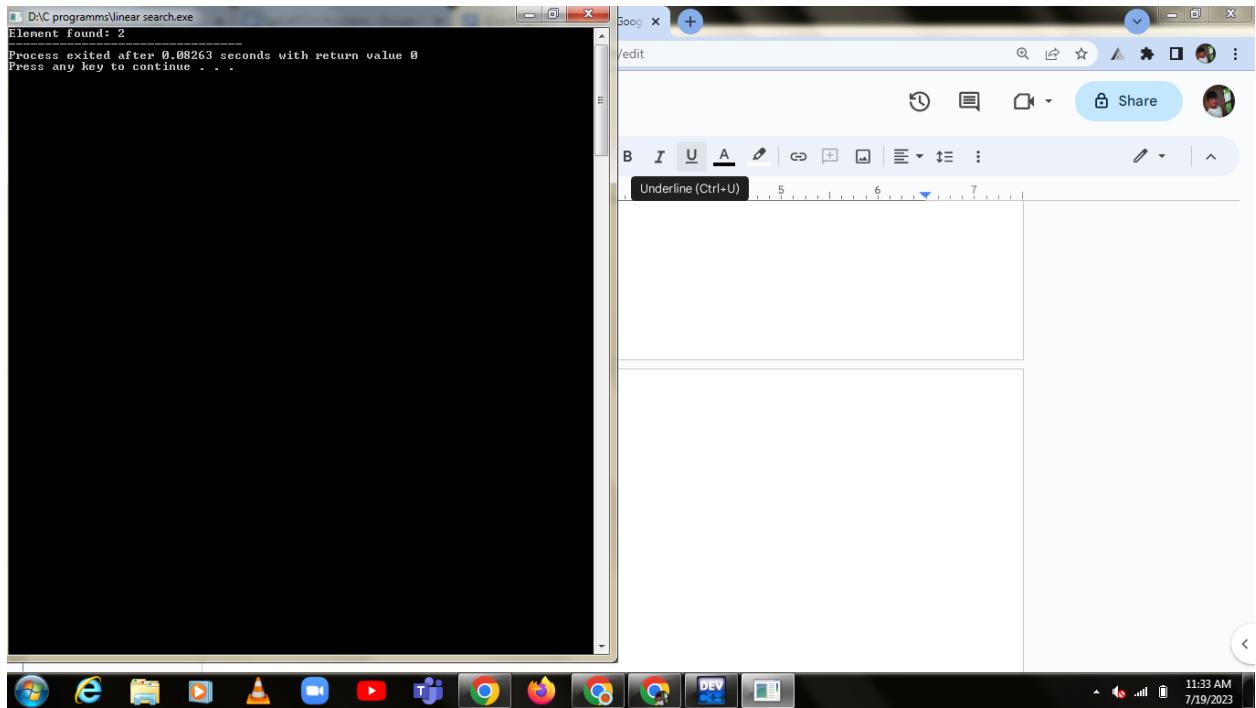
75.linear search

```
#include <iostream>
using namespace std;
int linearSearch(int arr[], int size, int x)
{
    for (int i = 0; i < size; i++)
    {
        if (arr[i] == x)
        {
            return i;
        }
    }
    return -1;
}
int main()
{
    int arr[] = {10, 20, 30, 40, 50};
    int x = 30;
    int n = sizeof(arr) / sizeof(arr[0]);
    int index = linearSearch(arr, n, x);
    if (index == -1)
    {
        cout << "Element not found";
    }
}
```

```

} else
{
    cout << "Element found: " << index;
}
return 0;
}

```



76.insertion sort

```

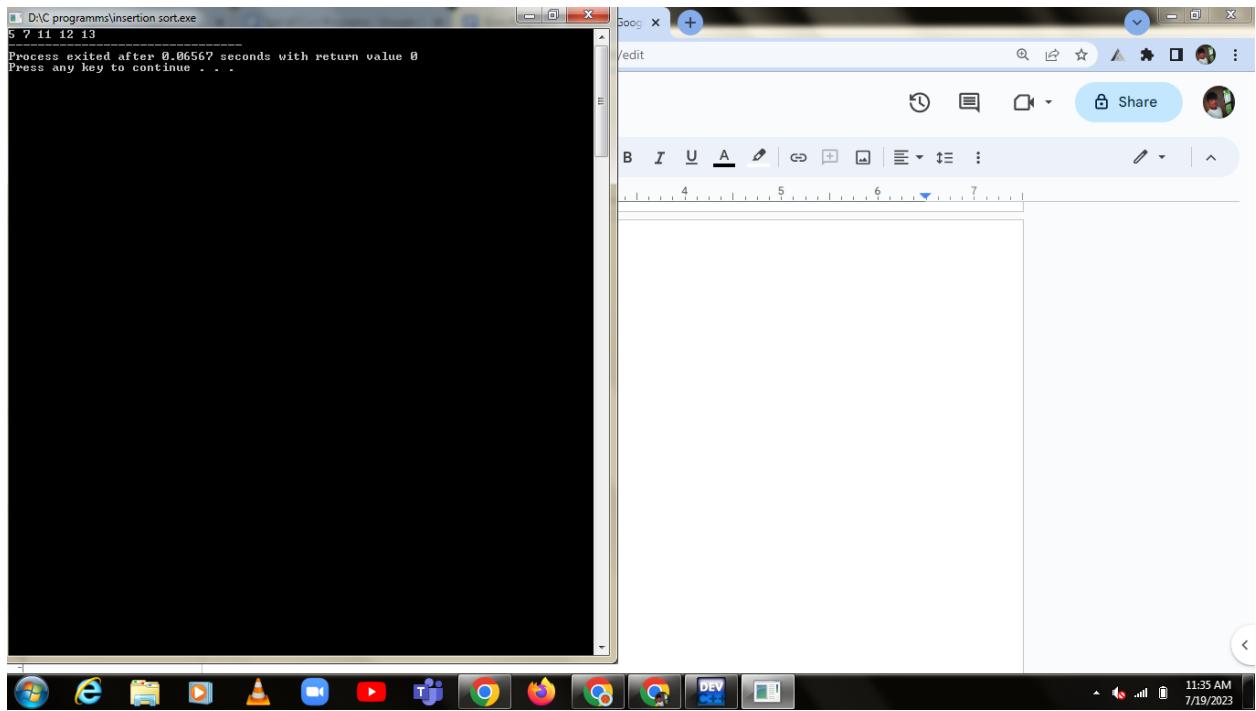
#include <iostream>
using namespace std;
void insertionSort(int arr[], int n)
{
    for (int i = 1; i < n; i++)
    {
        int key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j] > key)
        {
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = key;
    }
}

```

```

}
int main()
{
    int arr[] = {13,5,11,7,12};
    int n = sizeof(arr) / sizeof(arr[0]);
    insertionSort(arr, n);
    for (int i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
    return 0;
}

```



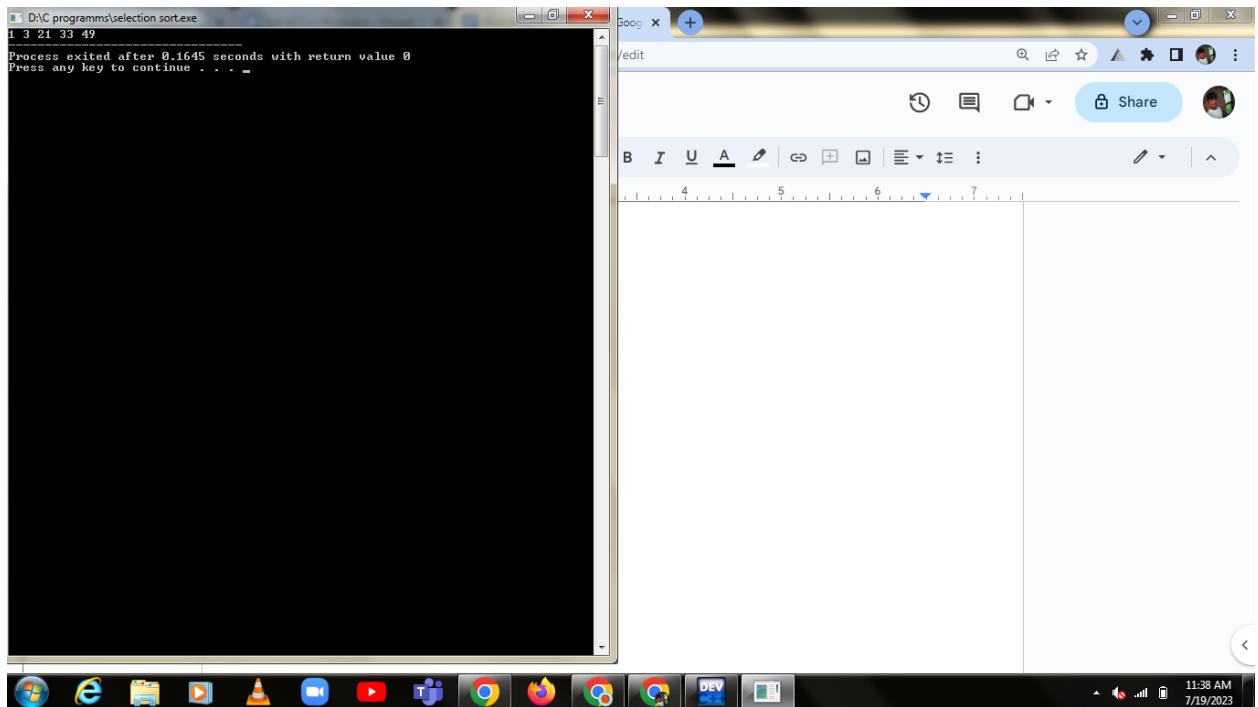
77.selection sort

```

#include <iostream>
using namespace std;
void selectionSort(int arr[], int n)
{
    for (int i = 0; i < n - 1; i++)
    {
        int minIndex = i;
        for (int j = i + 1; j < n; j++)
        {
            if (arr[j] < arr[minIndex])
                minIndex = j;
        }
        swap(arr[i], arr[minIndex]);
    }
}

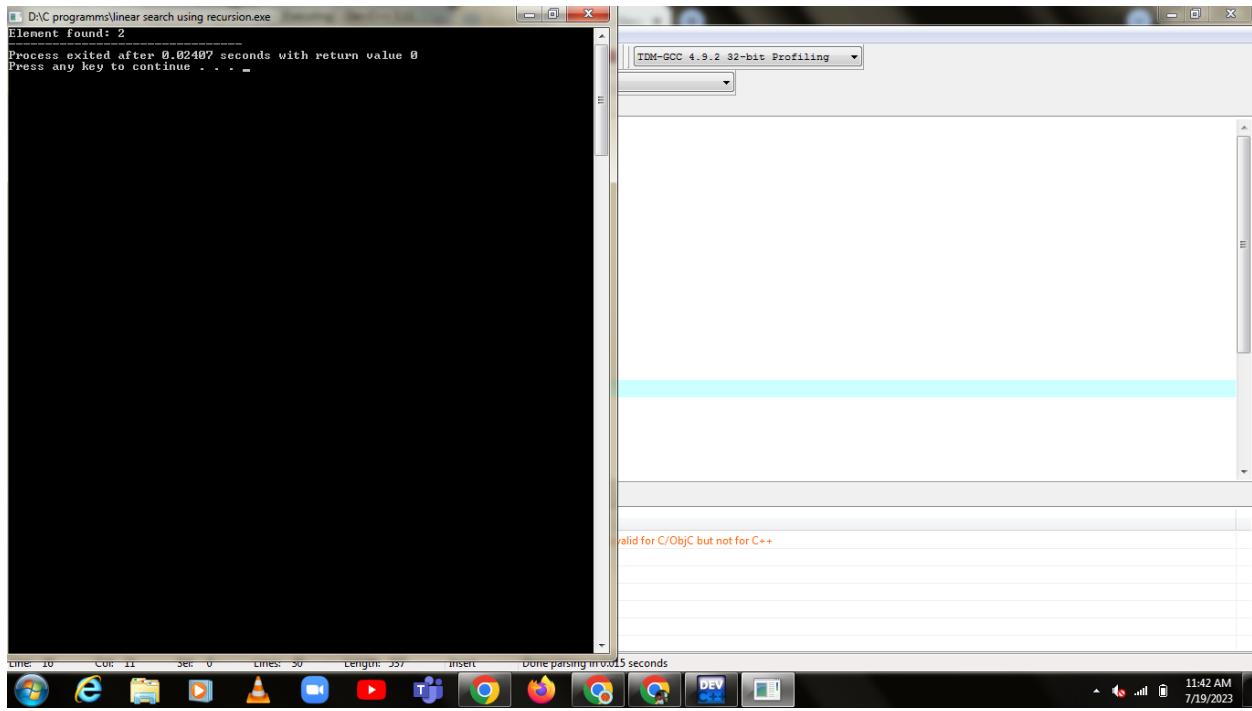
```

```
{  
if (arr[j] < arr[minIndex])  
{  
    minIndex = j;  
}  
}  
int temp = arr[i];  
arr[i] = arr[minIndex];  
arr[minIndex] = temp;  
}  
}  
int main()  
{  
    int arr[] = {49,3,21,33,1};  
    int n = sizeof(arr) / sizeof(arr[0]);  
    selectionSort(arr, n);  
    for (int i = 0; i < n; i++)  
    {  
        cout << arr[i] << " ";  
    }  
    return 0;  
}
```



79.linear search using recursion

```
#include <iostream>
using namespace std;
int linearSearch(int arr[], int size, int x)
{
    if (size == 0)
    {
        return -1;
    } else if (arr[0] == x)
    {
        return 0;
    } else
    {
        return linearSearch(arr + 1, size - 1, x);
    }
}
int main()
{
    int arr[] = {10, 20, 30, 40, 50};
    int x = 30;
    int n = sizeof(arr) / sizeof(arr[0]);
    int index = linearSearch(arr, n, x);
    if (index == -1)
    {
        cout << "Element not found";
    } else {
        cout << "Element found: " << index;
    }
    return 0;
}
```



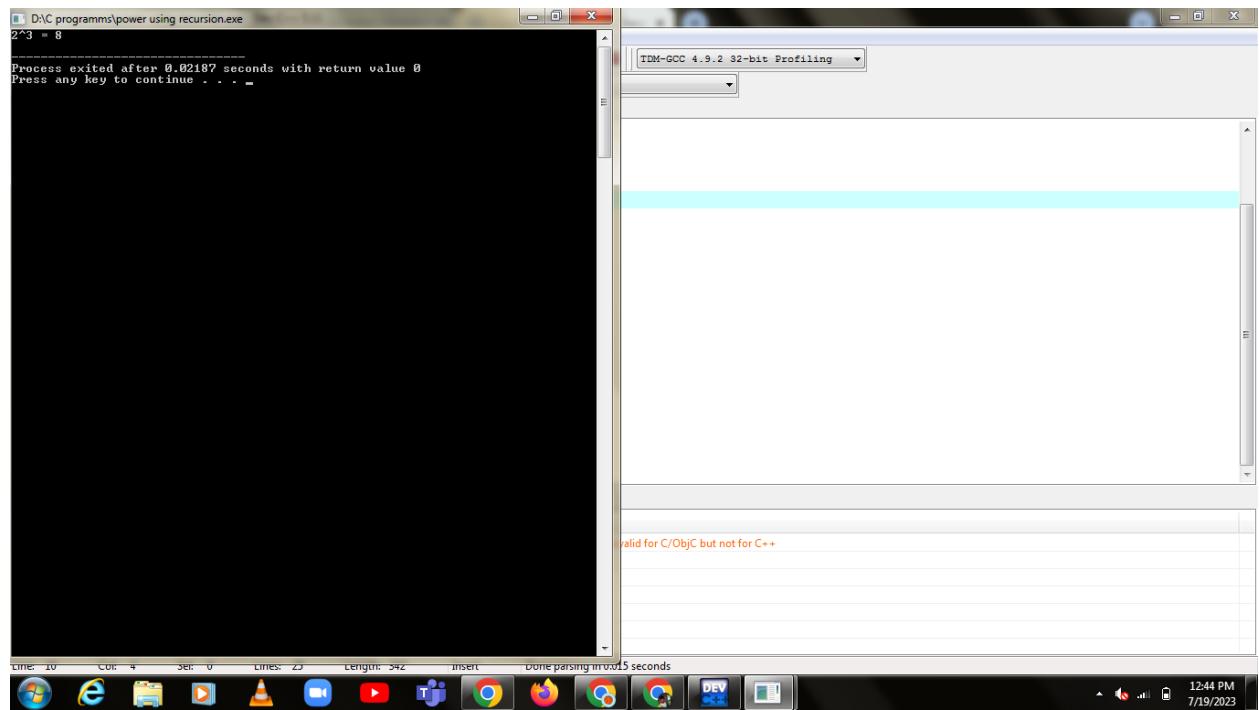
80.power using recursion

```
#include <iostream>
using namespace std;
int power(int x, int n)
{
    if (n == 0)
    {
        return 1;
    }
    else if (n == 1)
    {
        return x;
    } else
    {
        return x * power(x, n - 1);
    }
}
int main()
{
    int x = 2;
    int n = 3;
    int result = power(x, n);
```

```

cout << "2^3 = " << result << endl;
return 0;
}

```

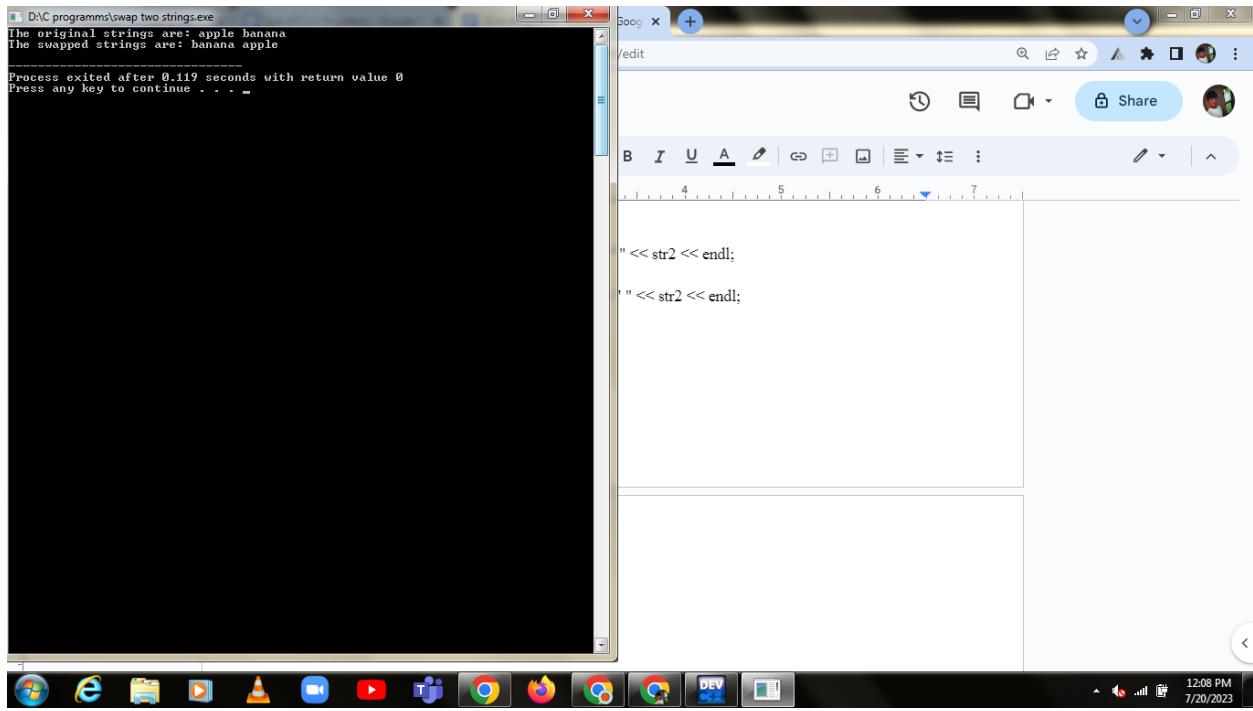


81.swap two strings

```

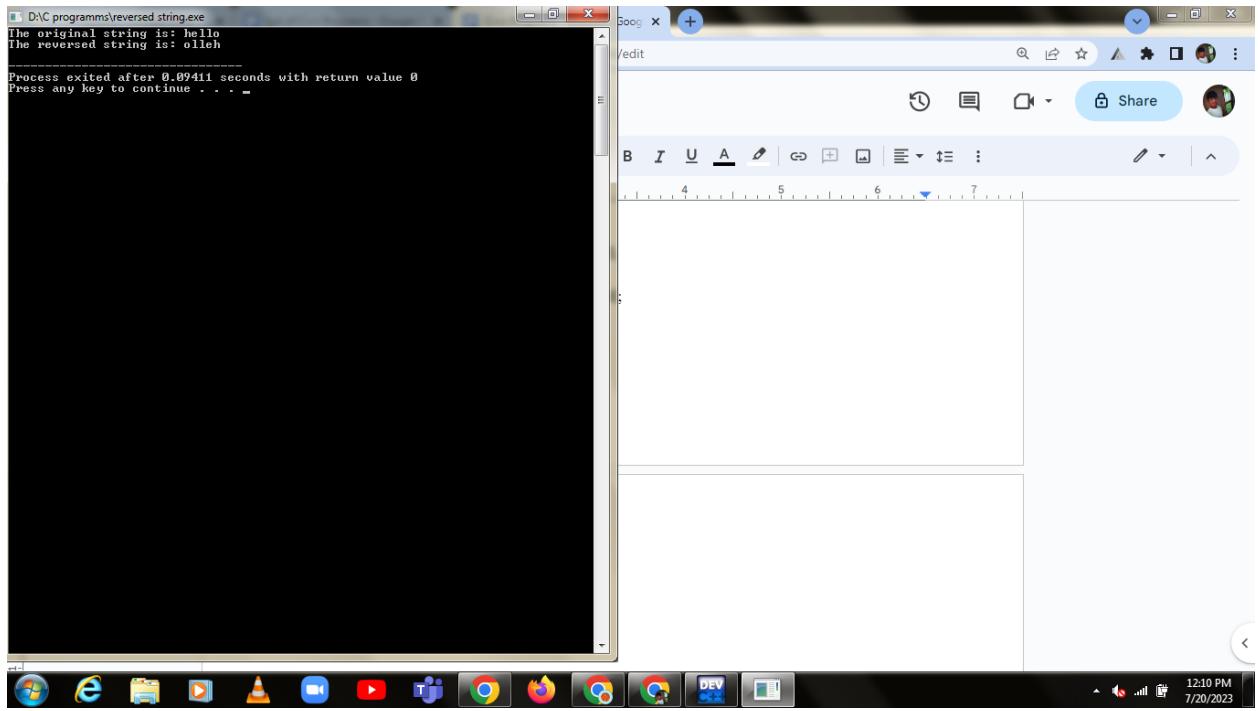
#include <iostream>
using namespace std;
void swapStrings(string &str1, string &str2)
{
    string temp = str1;
    str1 = str2;
    str2 = temp;
}
int main()
{
    string str1 = "apple";
    string str2 = "banana";
    cout << "The original strings are: " << str1 << " " << str2 << endl;
    swapStrings(str1, str2);
    cout << "The swapped strings are: " << str1 << " " << str2 << endl;
    return 0;
}

```



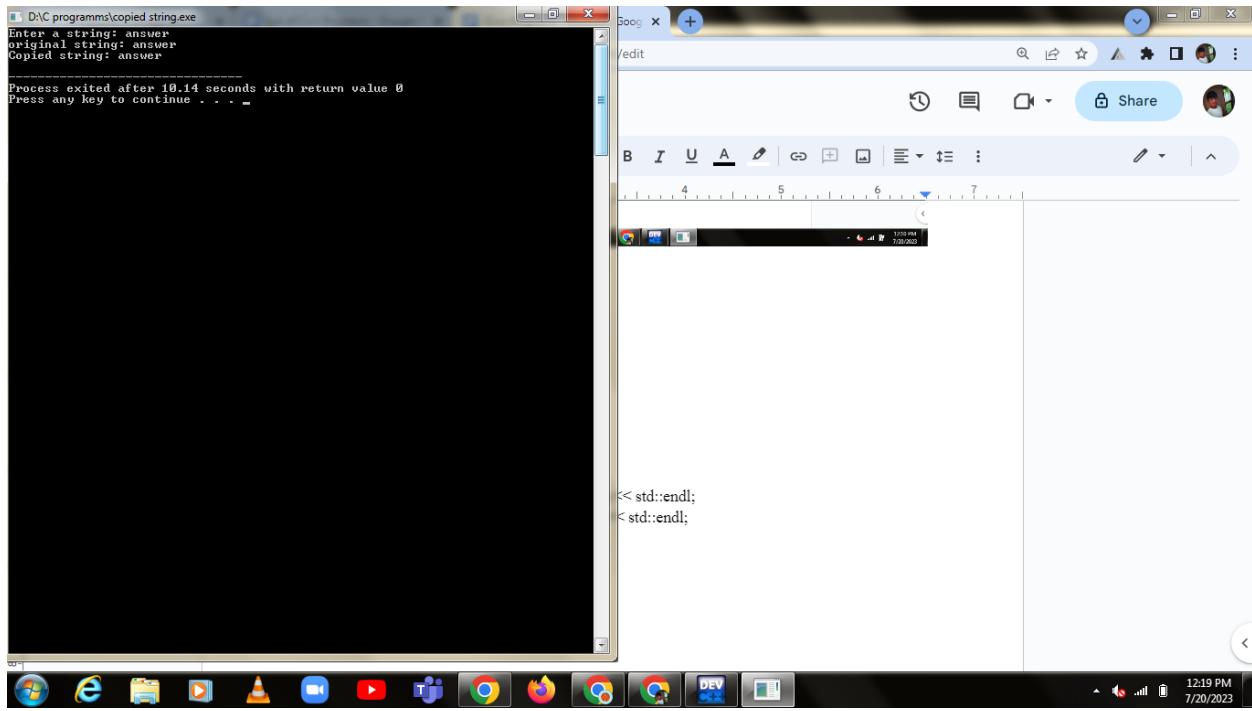
82.reversed string

```
#include <iostream>
#include <string>
using namespace std;
void reverseString(string &str)
{
    int n = str.length();
    for (int i = 0; i < n / 2; i++)
    {
        char temp = str[i];
        str[i] = str[n - i - 1];
        str[n - i - 1] = temp;
    }
}
int main()
{
    string str = "hello";
    cout << "The original string is: " << str << endl;
    reverseString(str);
    cout << "The reversed string is: " << str << endl;
    return 0;
}
```



83.copy string

```
#include <iostream>
#include <cstring>
int main() {
    const int MAX_LENGTH = 20;
    char source[MAX_LENGTH];
    char destination[MAX_LENGTH];
    std::cout << "Enter a string: ";
    std::cin.getline(source, MAX_LENGTH);
    std::strcpy(destination, source);
    std::cout << "original string: " << source << std::endl;
    std::cout << "Copied string: " << destination << std::endl;
    return 0;
}
```



85. find no of vowels,digits,consonants,spaces in a string

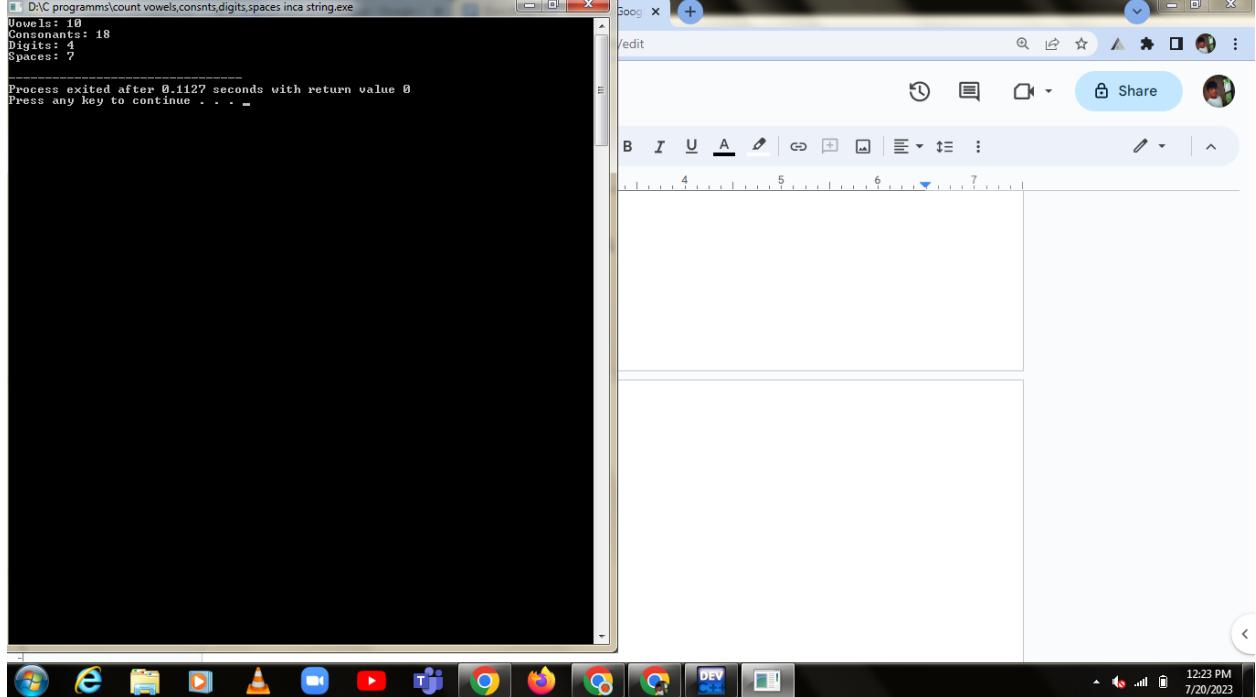
```
#include <iostream>
#include <string>
using namespace std;
int countVowels(string str)
{
    int vowels = 0;
    for (int i = 0; i < str.length(); i++)
    {
        if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u' ||
            str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U') {
            vowels++;
        }
    }
    return vowels;
}
int countConsonants(string str)
{
    int consonants = 0;
    for (int i = 0; i < str.length(); i++)
    {
        if (str[i] != 'a' && str[i] != 'e' && str[i] != 'i' && str[i] != 'o' && str[i] != 'u' &&
```

```

        str[i] != 'A' && str[i] != 'E' && str[i] != 'T' && str[i] != 'O' && str[i] != 'U' &&
        str[i] != ' ')
    {
        consonants++;
    }
}
return consonants;
}
int countDigits(string str)
{
    int digits = 0;
    for (int i = 0; i < str.length(); i++)
    {
        if (str[i] >= '0' && str[i] <= '9')
        {
            digits++;
        }
    }
    return digits;
}
int countSpaces(string str)
{
    int spaces = 0;
    for (int i = 0; i < str.length(); i++)
    {
        if (str[i] == ' ')
        {
            spaces++;
        }
    }
    return spaces;
}
int main() {
    string str = "i will eat 25 chocolates in 10 days";
    int vowels = countVowels(str);
    int consonants = countConsonants(str);
    int digits = countDigits(str);
    int spaces = countSpaces(str);
    cout << "Vowels: " << vowels << endl;
    cout << "Consonants: " << consonants << endl;
}

```

```
cout << "Digits: " << digits << endl;
cout << "Spaces: " << spaces << endl;
return 0;
}
```



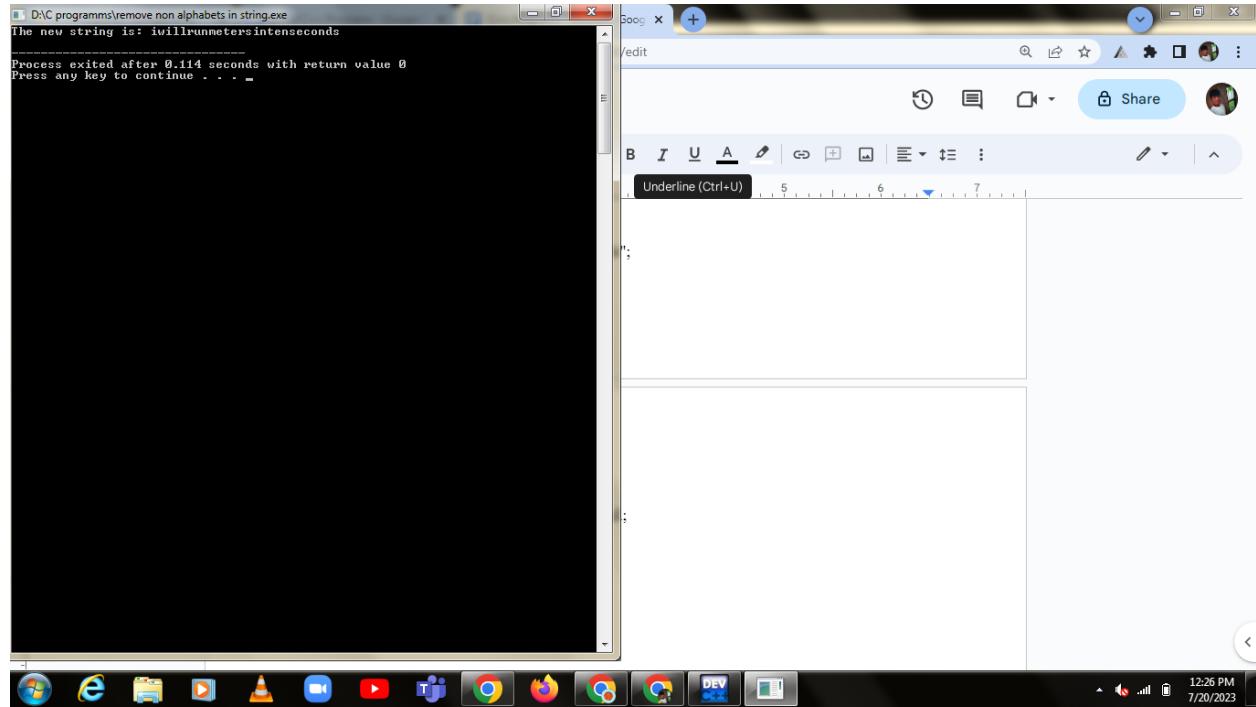
86.remove non alphabets in a string

```
#include <iostream>
#include <string>
using namespace std;
string removeNonAlpha(string str)
{
    string newStr = "";
    for (int i = 0; i < str.length(); i++)
    {
        if (isalpha(str[i]))
        {
            newStr += str[i];
        }
    }
    return newStr;
}
int main() {
    string str = "i will run 100 meters in ten seconds";
    cout << removeNonAlpha(str);
}
```

```

string newStr = removeNonAlpha(str);
cout << "The new string is: " << newStr << endl;
return 0;
}

```



88.print numbers from 1 to n using class

```

#include <iostream>
class Printnum
{
public:
    Printnum(int n) : num(n) {}
    void printNumbers()
    {
        if (num <= 0)
        {
            std::cout << "enter a number" << std::endl;
            return;
        }
        for (int i = 1; i <= num; i++)
        {
            std::cout << i << " ";
        }
        std::cout << std::endl;
    }
}

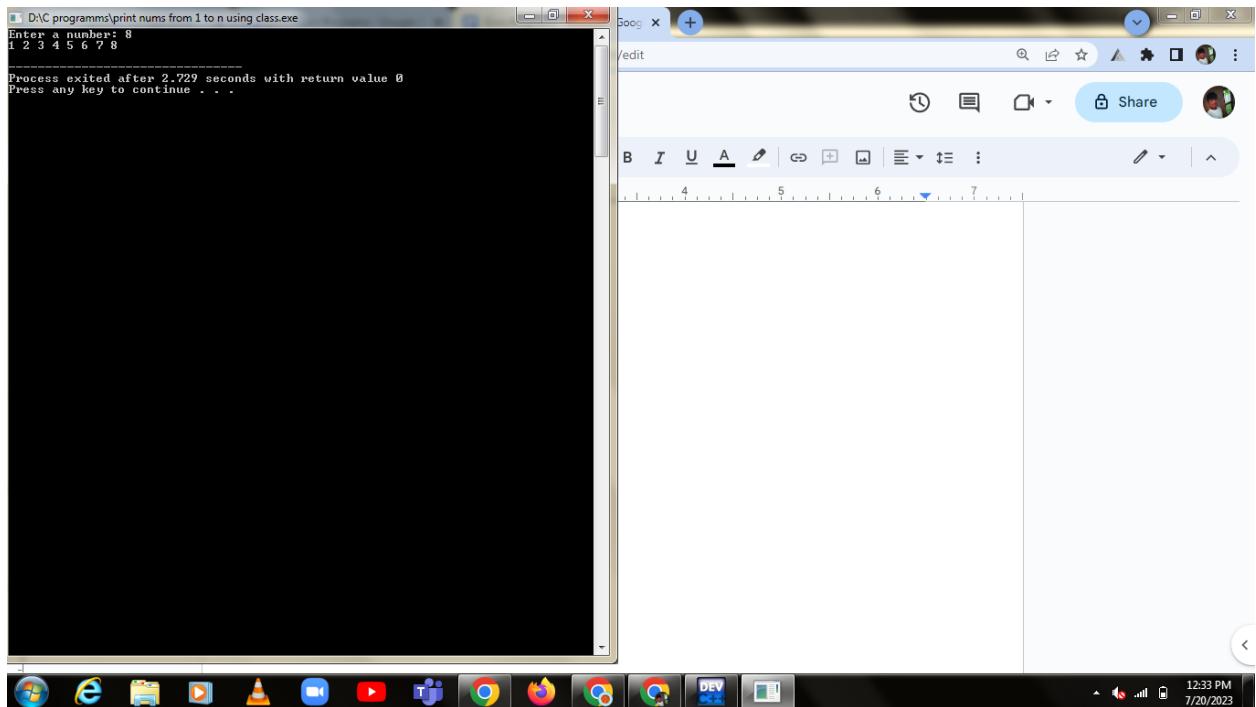
```

```

    }
private:
    int num;
};

int main()
{
    int n;
    std::cout << "Enter a number: ";
    std::cin >> n;
    Printnum np(n);
    np.printNumbers();
    return 0;
}

```



90.convert first letter of each word to uppercase and other to lowercase

```

#include <iostream>
#include <string>
using namespace std;
void capitalFirstLetter(string &str)
{
    for (int i = 0; i < str.length(); i++)
    {

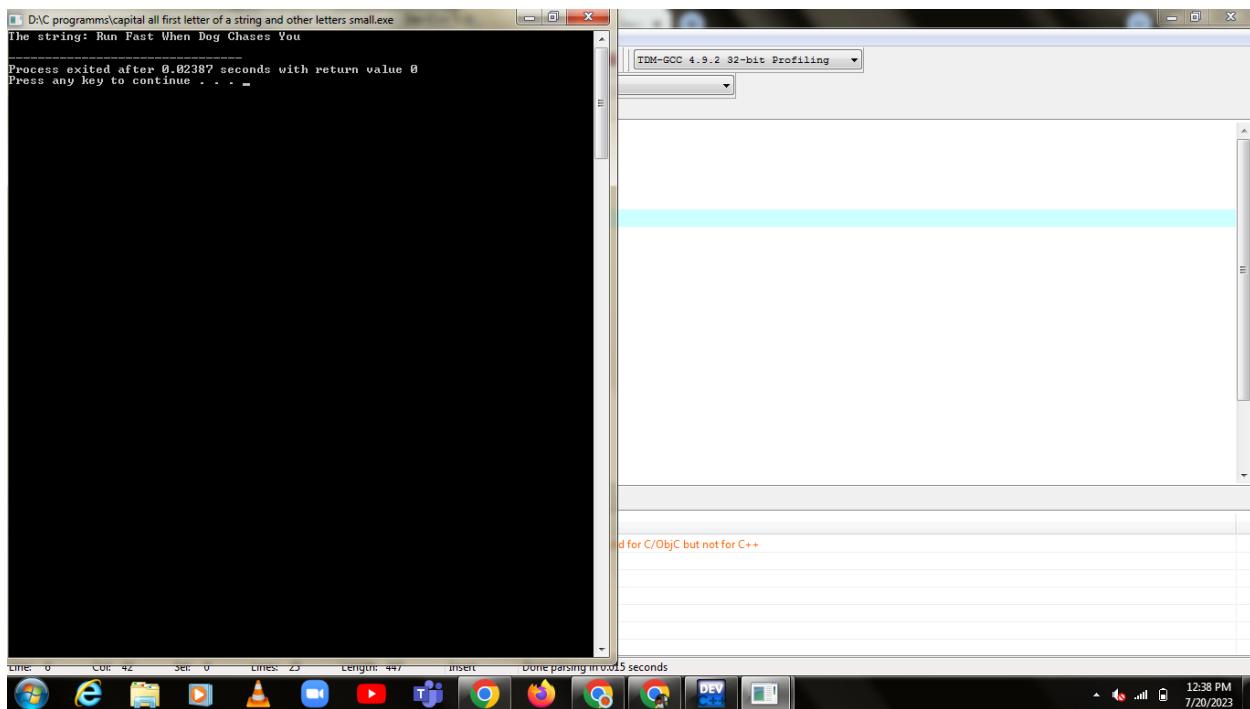
```

```

if (i == 0 || str[i - 1] == ' ')
{
    str[i] = toupper(str[i]);
}
else
{
    str[i] = tolower(str[i]);
}
}

int main()
{
    string str = "run fast when dog chases you";
    capitalFirstLetter(str);
    cout << "The string: " << str << endl;
    return 0;
}

```



91.octa to decimal

```

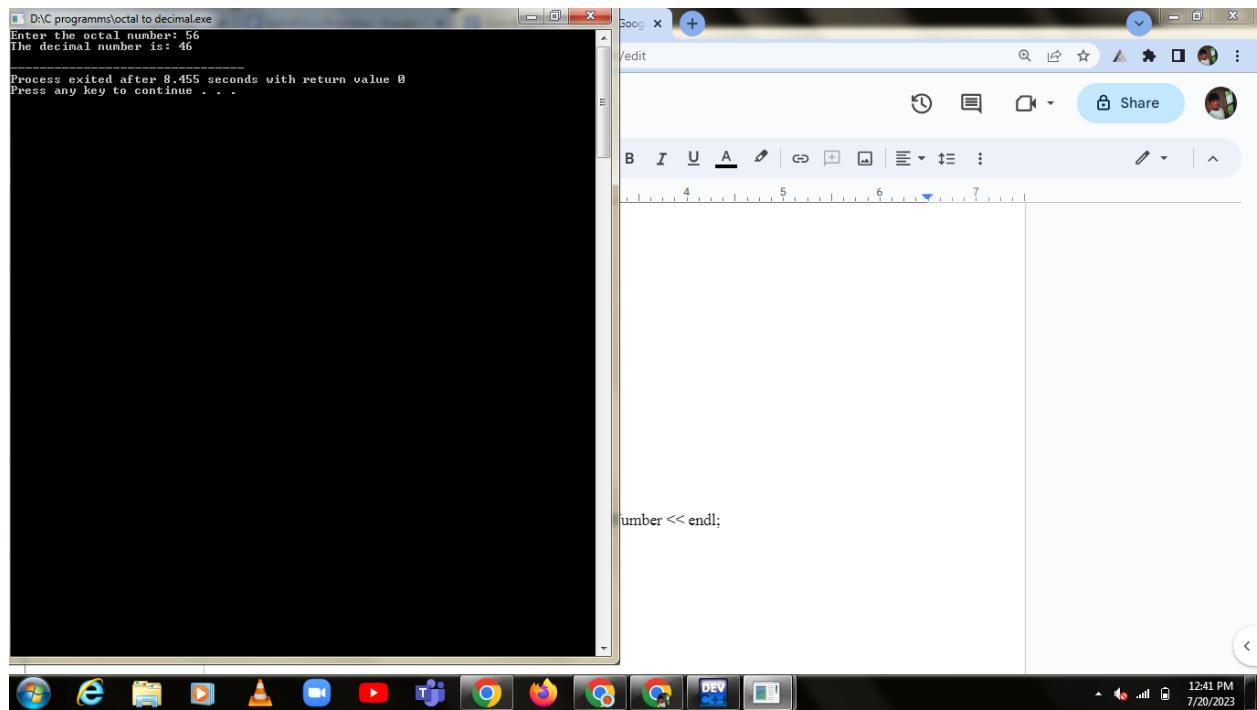
#include <iostream>
using namespace std;
int octalToDecimal(int n)
{

```

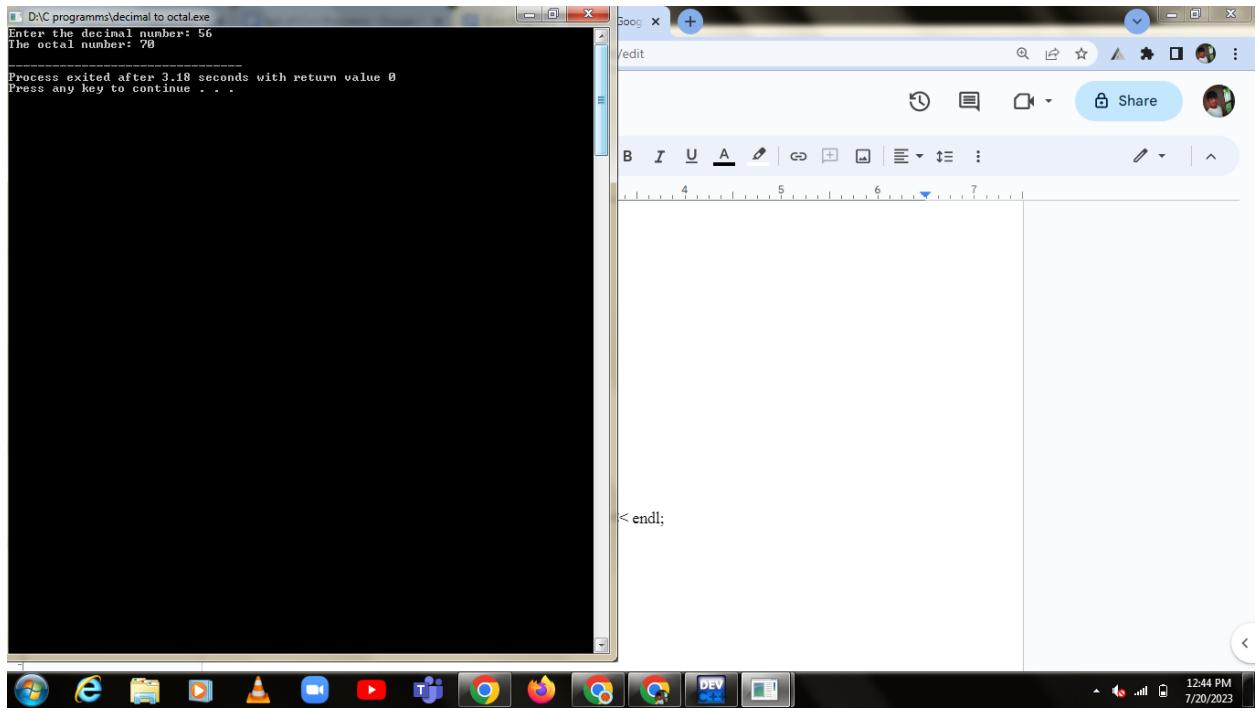
```

int decimalNumber = 0;
int base = 1;
while (n != 0) {
    int digit = n % 10;
    decimalNumber += digit * base;
    base *= 8;
    n /= 10;
}
return decimalNumber;
}
int main()
{
int n;
cout << "Enter the octal number: ";
cin >> n;
int decimalNumber = octalToDecimal(n);
cout << "The decimal number is: " << decimalNumber << endl;
return 0;
}

```

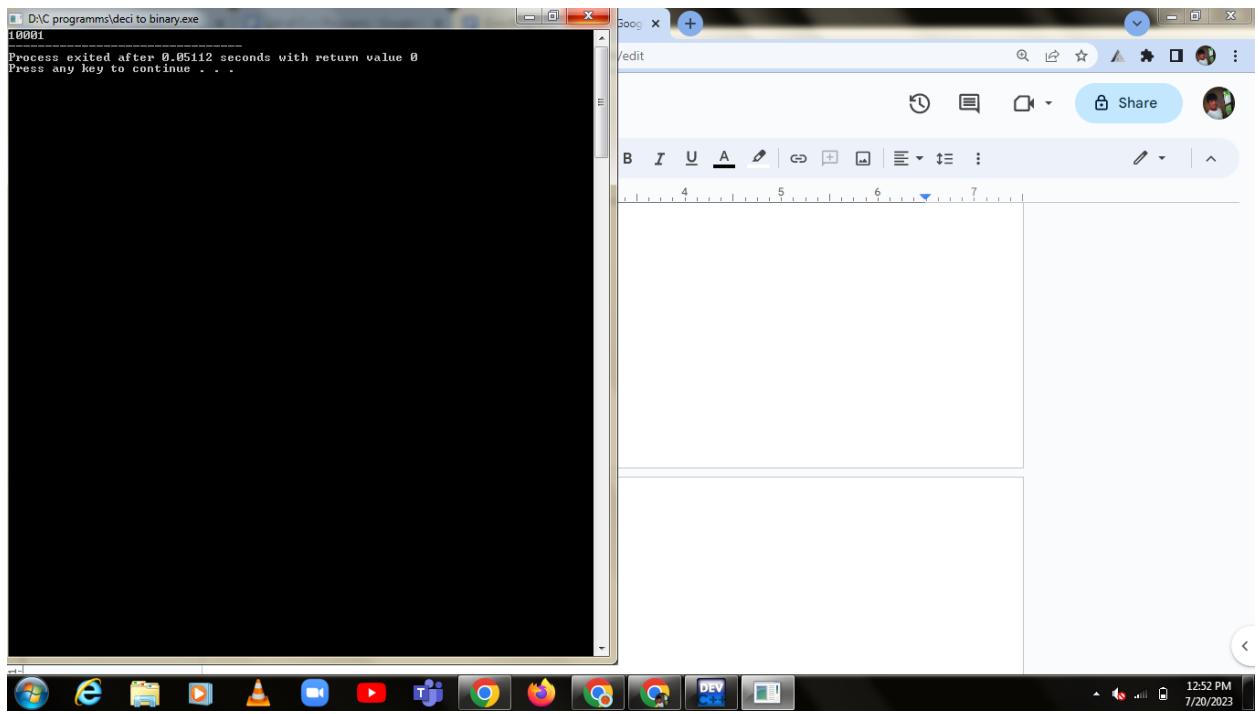


```
92.decimal to octal
#include <iostream>
using namespace std;
int decimalToOctal(int n)
{
    int octalNumber = 0;
    int i = 1;
    while (n > 0)
    {
        int remainder = n % 8;
        octalNumber += remainder * i;
        i *= 10;
        n /= 8;
    }
    return octalNumber;
}
int main() {
    int n;
    cout << "Enter the decimal number: ";
    cin >> n;
    int octalNumber = decimalToOctal(n);
    cout << "The octal number: " << octalNumber << endl;
    return 0;
}
```



94.decimal to binary

```
#include <iostream>
using namespace std;
void decToBinary(int n)
{
    int binaryNum[32];
    int i = 0;
    while (n > 0)
    {
        binaryNum[i] = n % 2;
        n = n / 2;
        i++;
    }
    for (int j = i - 1; j >= 0; j--)
        cout << binaryNum[j];
}
int main()
{
    int n = 17;
    decToBinary(n);
    return 0;
}
```



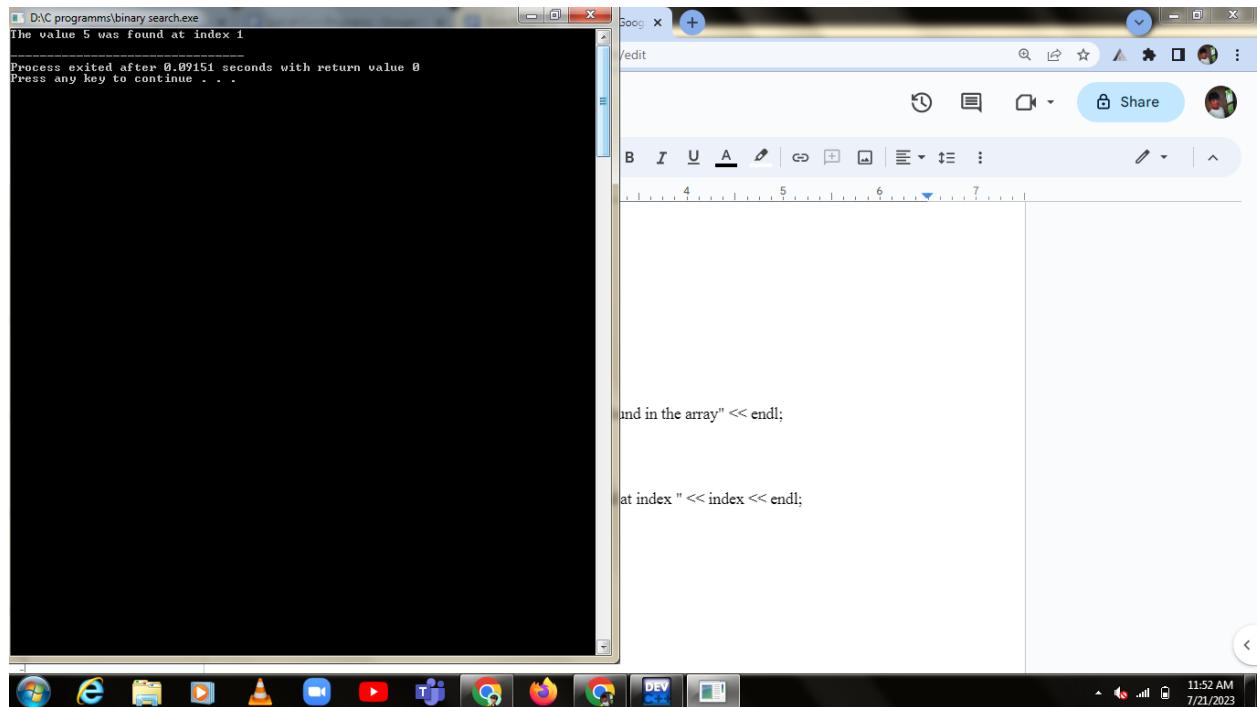
95.binary search

```
#include <iostream>
using namespace std;
int binary_search(int *array, int size, int value)
{
    int low = 0;
    int high = size - 1;
    int mid;
    while (low <= high)
    {
        mid = (low + high) / 2;
        if (array[mid] == value)
        {
            return mid;
        }
        else if (array[mid] < value)
        {
            low = mid + 1;
        }
        else
        {
```

```

        high = mid - 1;
    }
}
return -1;
}
int main()
{
    int array[] = {10, 5, 2, 1, 8, 9, 7, 6, 4, 3};
    int size = sizeof(array) / sizeof(array[0]);
    int value = 5;
    int index = binary_search(array, size, value);
    if (index == -1)
    {
        cout << "The value " << value << " was not found in the array" << endl;
    }
    else
    {
        cout << "The value " << value << " was found at index " << index << endl;
    }
    return 0;
}

```



97.sum of diagonals of matrix

```
#include <iostream>
using namespace std;
int main()
{
    int n;
    cout << "Enter the size of the matrix: ";
    cin >> n;
    int matrix[n][n];
    cout << "Enter the elements of the matrix: " << endl;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cin >> matrix[i][j];
        }
    }
    int sum_of_diagonals = 0;
    for (int i = 0; i < n; i++)
    {
        sum_of_diagonals += matrix[i][i];
        sum_of_diagonals += matrix[i][n - i - 1];
    }
    cout << "The sum of diagonals is " << sum_of_diagonals << endl;
    return 0;
}
```

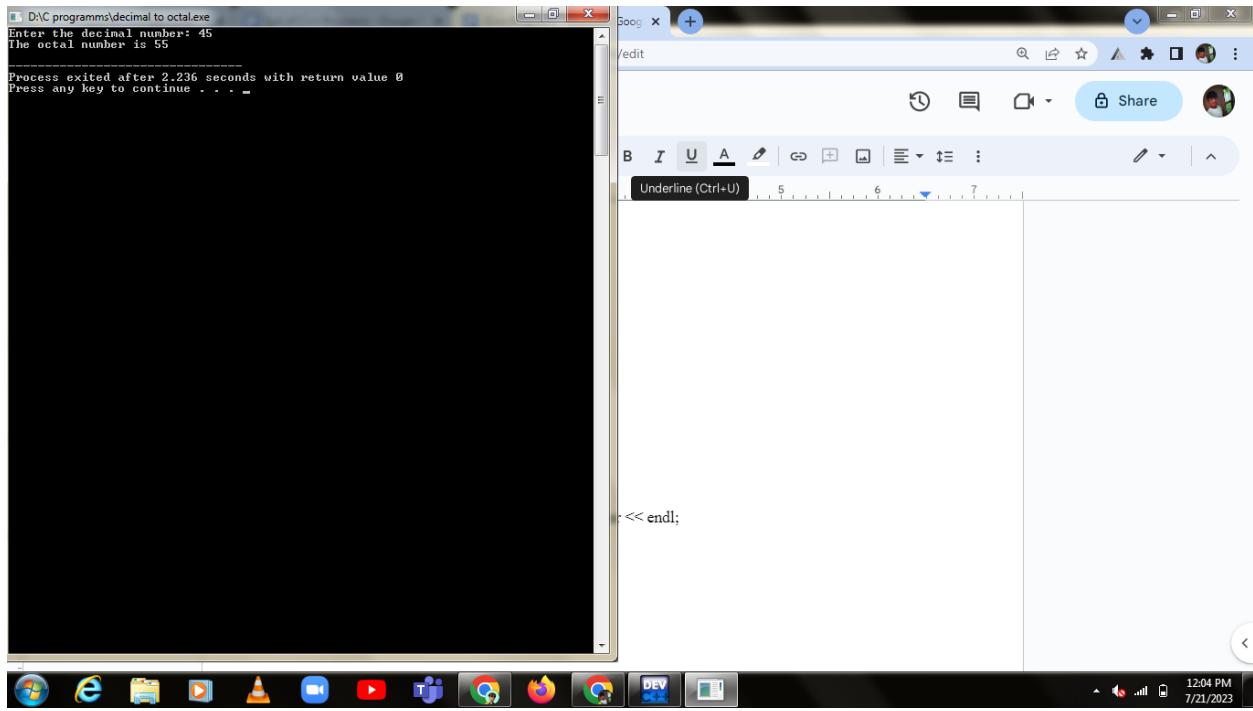
A screenshot of a Windows desktop environment. On the left, a black terminal window titled 'D:\C programs\sum of diagonals of matrix.exe' displays the output of a C++ program. The program asks for the size of a matrix (3x3), takes input elements (1, 2, 3, 4, 5, 6, 7, 8, 9), calculates the sum of diagonals (30), and exits. On the right, a browser window titled 'edit' is open, showing a blank page with a toolbar at the top. The taskbar at the bottom shows various pinned icons and the system clock indicating 11:58 AM on 7/21/2023.

```
D:\C programs\sum of diagonals of matrix.exe
Enter the size of the matrix: 3
Enter the elements of the matrix:
1
2
3
4
5
6
7
8
9
The sum of diagonals is 30
Process exited after 11.22 seconds with return value 0
Press any key to continue . . .

Diagonals << endl;
```

99.decimal to octal

```
#include <iostream>
using namespace std;
int main()
{
    int decimal_number;
    cout << "Enter the decimal number: ";
    cin >> decimal_number;
    int octal_number = 0;
    int i = 1;
    while (decimal_number > 0)
    {
        int remainder = decimal_number % 8;
        octal_number += remainder * i;
        i *= 10;
        decimal_number /= 8;
    }
    cout << "The octal number is " << octal_number << endl;
    return 0;
}
```



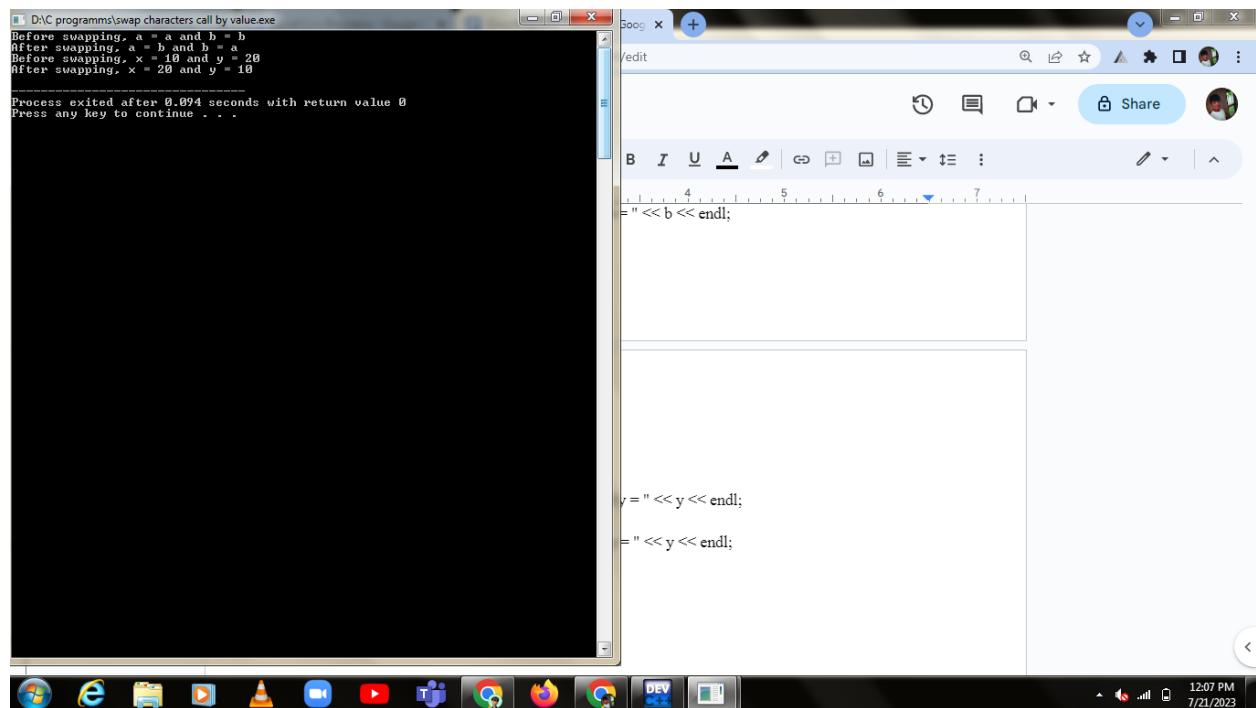
100.swap characters and integers call by value

```
#include <iostream>
using namespace std;
void swap_characters(char *a, char *b)
{
    char temp = *a;
    *a = *b;
    *b = temp;
}
void swap_integers(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}
int main()
{
    char a = 'a';
    char b = 'b';
    cout << "Before swapping, a = " << a << " and b = " << b << endl;
    swap_characters(&a, &b);
    cout << "After swapping, a = " << a << " and b = " << b << endl;
```

```

int x = 10;
int y = 20;
cout << "Before swapping, x = " << x << " and y = " << y << endl;
swap_integers(&x, &y);
cout << "After swapping, x = " << x << " and y = " << y << endl;
return 0;
}

```



101.reverse of number by defining function outside class

```

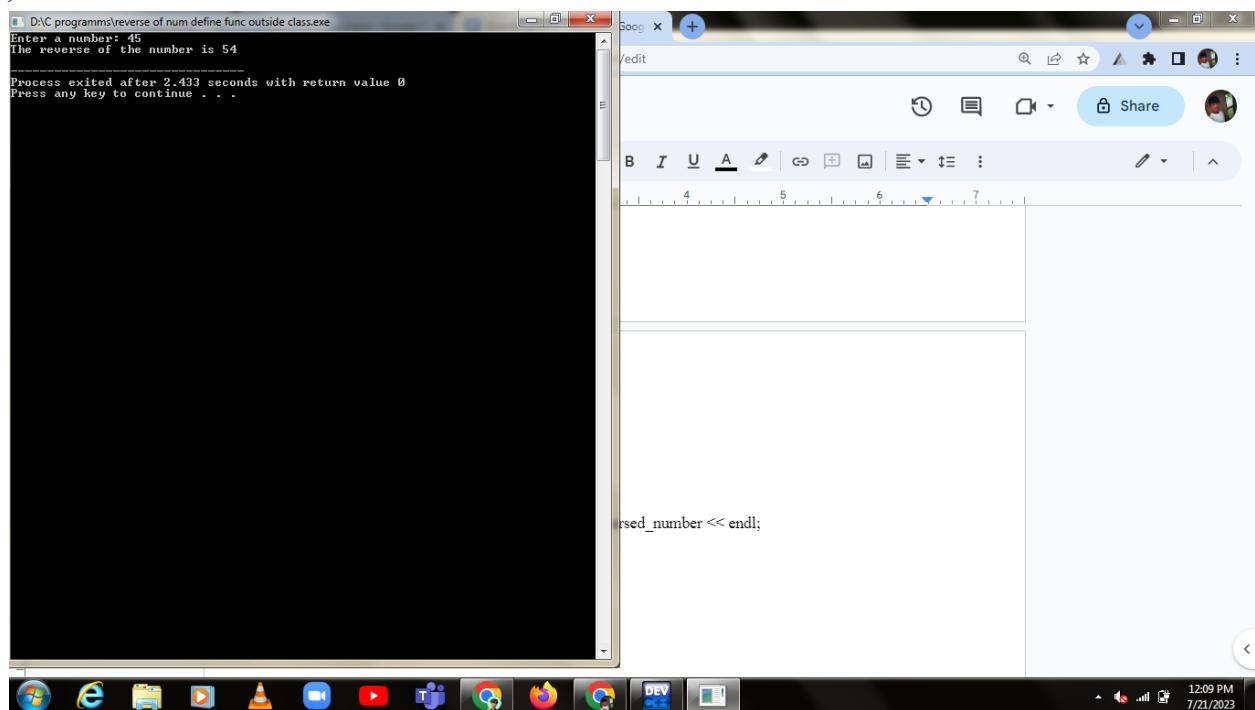
#include <iostream>
using namespace std;
int reverse(int number)
{
    int reversed_number = 0;
    while (number > 0)
    {
        int digit = number % 10;
        reversed_number = reversed_number * 10 + digit;
        number /= 10;
    }
    return reversed_number;
}
int main()
{

```

```

int number;
cout << "Enter a number: ";
cin >> number;
int reversed_number = reverse(number);
cout << "The reverse of the number is " << reversed_number << endl;
return 0;
}

```



102.swap 2 char and int call by address

```

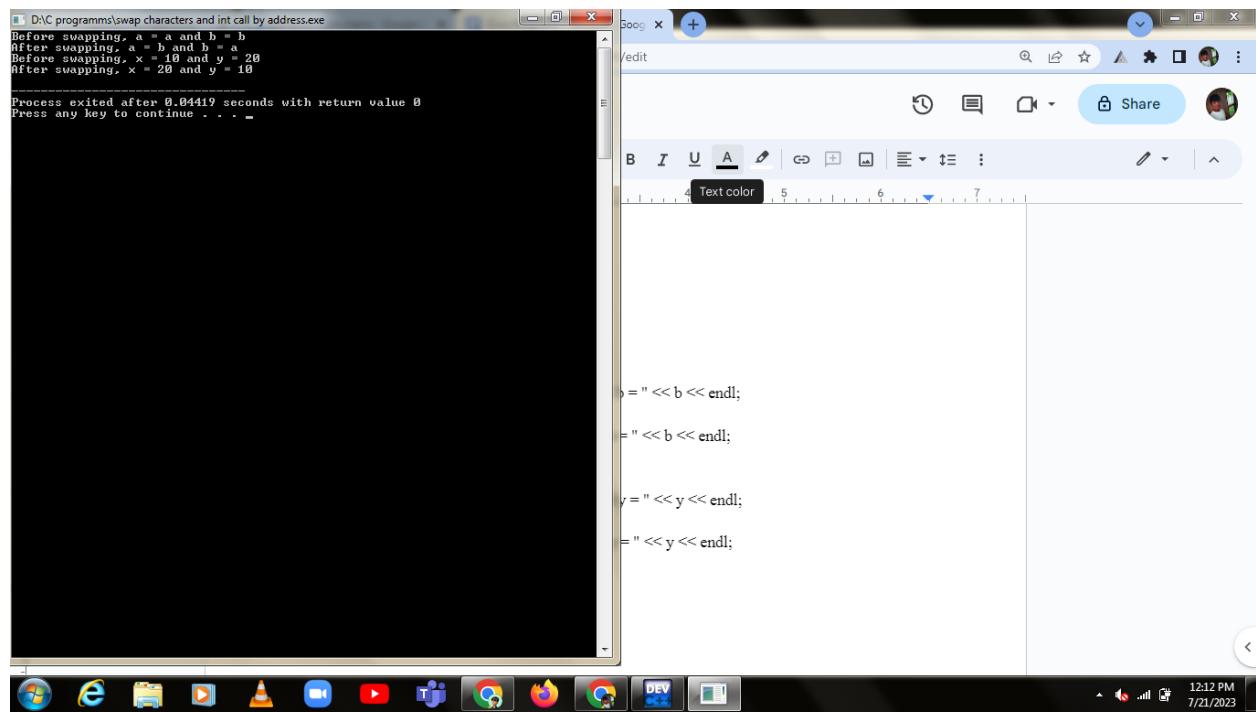
#include <iostream>
using namespace std;
void swap_characters_by_address(char *a, char *b)
{
    char temp = *a;
    *a = *b;
    *b = temp;
}
void swap_integers_by_address(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

```

```

int main()
{
    char a = 'a';
    char b = 'b';
    cout << "Before swapping, a = " << a << " and b = " << b << endl;
    swap_characters_by_address(&a, &b);
    cout << "After swapping, a = " << a << " and b = " << b << endl;
    int x = 10;
    int y = 20;
    cout << "Before swapping, x = " << x << " and y = " << y << endl;
    swap_integers_by_address(&x, &y);
    cout << "After swapping, x = " << x << " and y = " << y << endl;
    return 0;
}

```



103.inheritance beyond single level

```

#include <iostream>
using namespace std;
class Animal
{
protected:
    string name;
public:

```

```
Animal(string name)
{
    this->name = name;
}
void speak()
{
    cout << "I am an animal" << endl;
}
};

class Dog : public Animal
{
public:
    Dog(string name) : Animal(name) {}
    void bark() {
        cout << "worw" << endl;
    }
};

class Cat : public Animal
{
public:
    Cat(string name) : Animal(name) {}
    void meow()
    {
        cout << "Meow" << endl;
    }
};

int main()
{
    Dog dog("jimmy");
    dog.speak();
    dog.bark();
    Cat cat("timmy");
    cat.speak();
    cat.meow();
    return 0;
}
```

