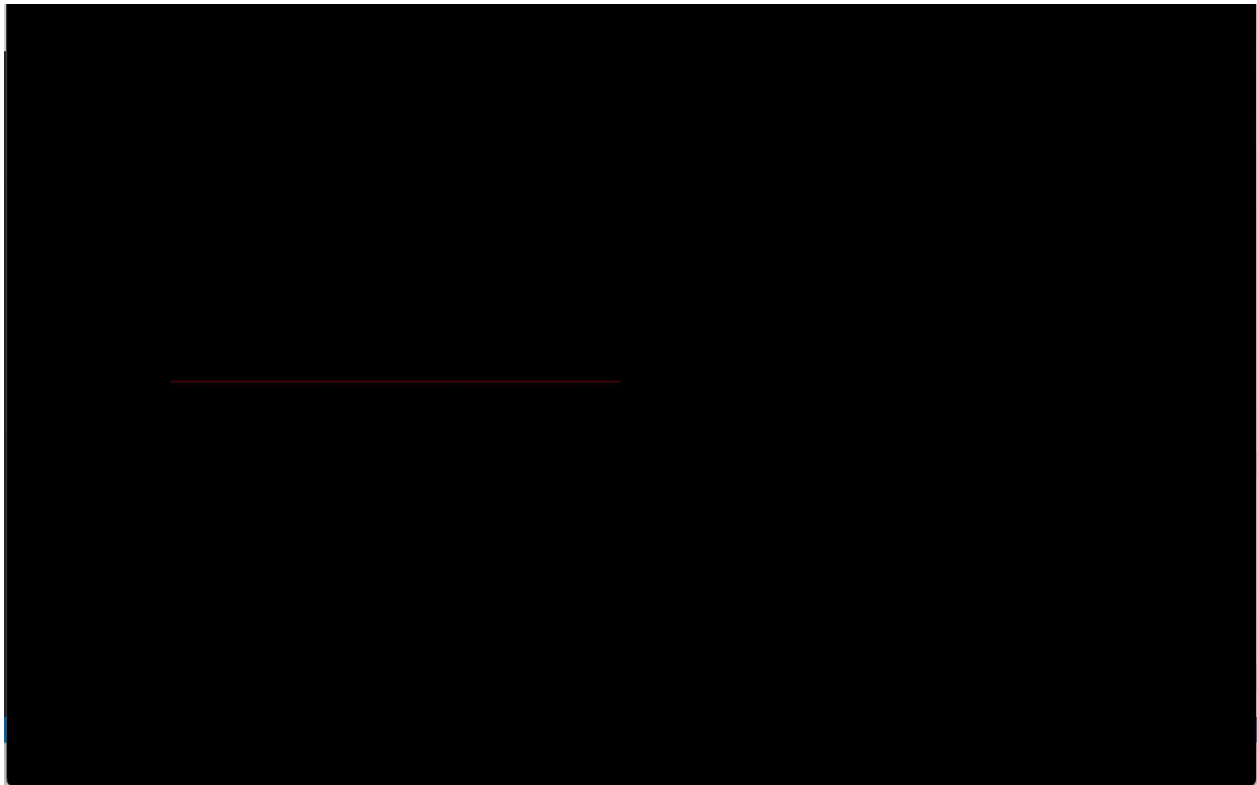


Program 1:- Draw Line Using DDA Algorithm

```
/*  
  
computer graphics practical  
  
Author: Sachin Parihar  
*/  
  
#include<stdio.h>  
#include<conio.h>  
#include<graphics.h>  
#include<math.h>  
  
void Line_dda(int x_a,int x_b,int y_a, int y_b){  
    int dx= x_b-x_a;  
    int dy=y_b-y_a;  
    int steps;  
    float x_increment,y_increment,x=x_a,y=y_a;  
    if(abs(dx)>abs(dy)){  
        steps=abs(dx);  
    }  
    else{ steps=abs(dy);}  
    x_increment=dx/(float)steps;  
    y_increment=dy/(float)steps;  
    putpixel(round(x),round(y),RED);  
    for(int k=0;k<steps;k++){  
        x +=x_increment;  
        y +=y_increment;  
        putpixel(round(x),round(y),RED);  
    }  
}
```

```
}  
int main(){  
    int gd= DETECT,gm;  
    initgraph(&gd,&gm, NULL);  
    initwindow(getmaxwidth(),getmaxheight());  
    Line_dda(200, 750,getmaxheight()/2,getmaxheight()/2);  
    getch();  
    closegraph();  
    return 0;  
}
```

OUTPUT:-



Program 2:-Draw line using Bresenham's Algorithm

```
/*  
  
computer graphics practical  
Author:Sachin Parihar  
*/  
  
#include<stdio.h>  
#include<conio.h>  
#include<graphics.h>  
  
void line_bres(int x_a,int y_a,int x_b,int y_b){  
    int dx=abs(x_a-x_b),dy=abs(y_a-y_b);  
    int p= 2*(dy-dx);  
    int twody = 2*dy;  
    int twodydx = 2*(dy-dx);  
    int x,y,xEnd;  
    if(x_a>x_b){  
        x=x_b;  
        y=y_b;  
        xEnd=x_a;  
    }  
    else{  
        x=x_a;  
        y=y_b;  
        xEnd=x_b;  
    }  
}
```

```

    }
    putpixel(x,y,GREEN);
    while(x<xEnd){
        x++;
        if(p<0){
            p +=twody;
        }
        else{
            y++;
            p +=twodydx;
        }
        putpixel(x,y,GREEN);
    }
}

int main(){
    int gd= DETECT,gm;
    initgraph(&gd,&gm,NULL);
    initwindow(getmaxwidth(),getmaxheight());
    line_bres(100,getmaxheight()/2,900,getmaxheight()/2);
    getch();
    closegraph();
    return 0;
}

```

OUTPUT : -



Program 3:-Draw circle using Polynomial Method

```
/*  
computer graphics practical  
Author:Sachin Parihar  
*/  
  
#include<stdio.h>  
#include<conio.h>  
#include<graphics.h>  
#include<math.h>  
  
void setPixel(int x,int y,int h, int k){  
    putpixel(x+h,y+k,GREEN);  
    putpixel(x+h,-y+k,GREEN);  
    putpixel(-x+h,-y+k,GREEN);  
    putpixel(-x+h,y+k,GREEN);  
    putpixel(y+h,x+k,GREEN);  
    putpixel(y+h,-x+k,GREEN);  
    putpixel(-y+h,-x+k,GREEN);  
}
```

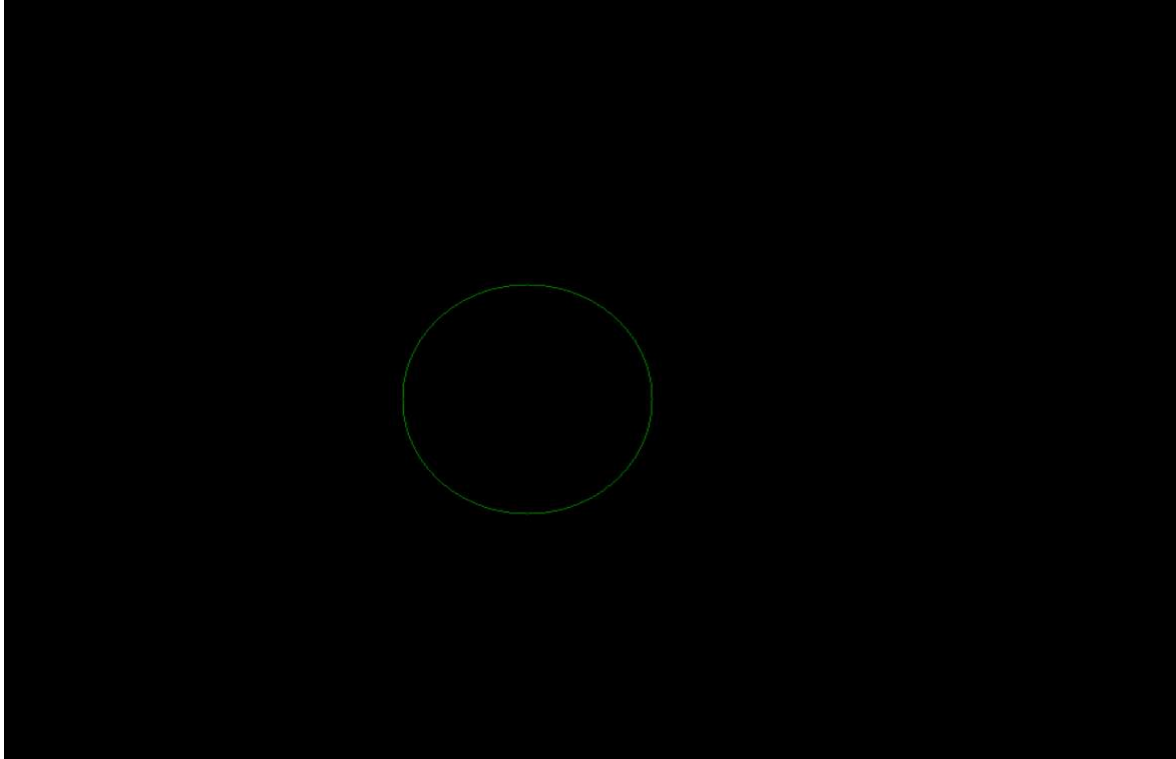
```

        putpixel(-y+h,x+k, GREEN);
    }
int main(){
    int gd= DETECT, gm;
    int x,y,r,i,h,k;
    double x2;
    initgraph(&gd,&gm, NULL);
    initwindow(getmaxwidth(),getmaxheight());
    h=getmaxwidth()/2;
    k=getmaxheight()/2;
    r=100;
    x=0;
    y=r; x2=r/sqrt(2);

    while(x<=x2){
        y=sqrt(r*r-x*x);
        setPixel(floor(x), floor(y), h, k);
        setPixel(floor(y), floor(x), h, k);
        setPixel(floor(-x), floor(-y), h, k);
        setPixel(floor(-y), floor(-x), h, k);
        x++;
    }
    getch();
    closegraph();
    return 0;}

```

OUTPUT: -

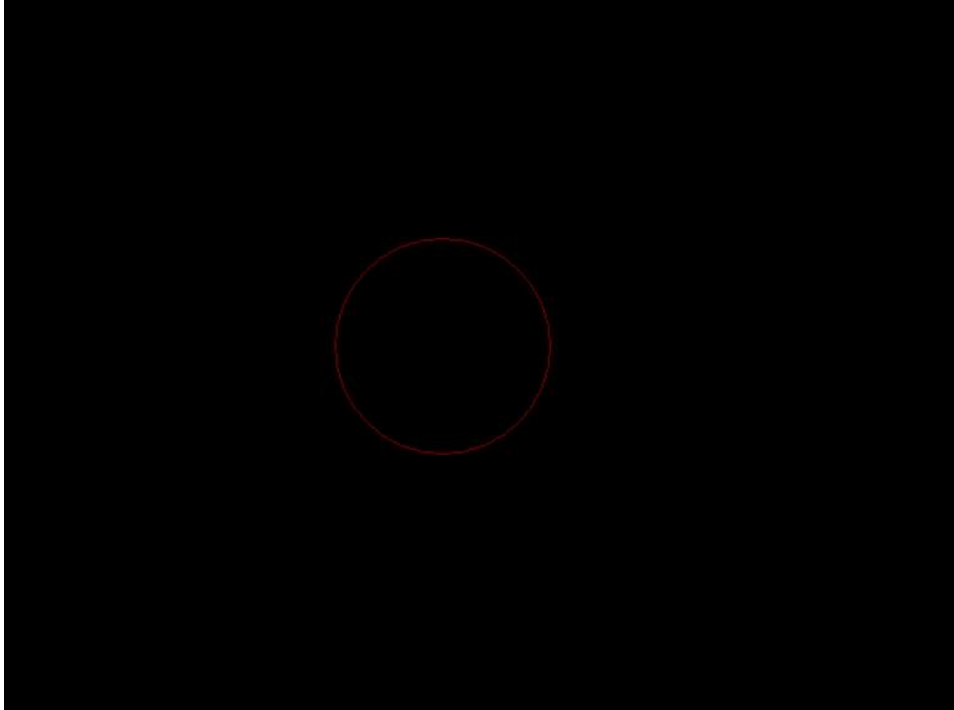


Program 4:- Draw circle using Trigonometric method

```
/*  
computer graphics practical  
Author: Sachin Parihar  
*/  
  
#include<stdio.h>  
#include<conio.h>  
#include<graphics.h>  
#include<math.h>  
  
void Circle(int x,int y,int radius){  
    for(int theta=0;theta<360;theta++){  
        float angle= theta*(M_PI)/180;
```

```
        x=x+radius*cos(angle);  
        y=y+radius*sin(angle);  
        putpixel(x,y,RED);  
    }  
  
}  
  
int main(){  
    int gd= DETECT,gm;  
    initgraph(&gd,&gm,NULL);  
    initwindow(getmaxwidth(),getmaxheight());  
    Circle(getmaxwidth()/2,getmaxheight()/2,3);  
    getch();  
    closegraph();  
    return 0;  
}
```

OUTPUT: -



Program 5:-Draw circle using Bresenham`s algorithm

/*

computer graphics practical

Author:Sachin Parihar

***/**

`#include<stdio.h>`

`#include<graphics.h>`

`void drawCircle(int xc,int yc,int x,int y){`

`putpixel(xc+x,yc+y,RED);`

`putpixel(xc-x,yc+y,RED);`

`putpixel(xc+x,yc-y,RED);`

`putpixel(xc-x,yc-y,RED);`

`putpixel(xc+y,yc+x,RED);`

`putpixel(xc-y,yc+x,RED);`

`putpixel(xc+y,yc-x,RED);`

`putpixel(xc-y,yc-x,RED);`

`}`

`void Circle(int xc,int yc,int r){`

`int x=0,y=r;`

`int d=3-2*r;`

`drawCircle(xc,yc,x,y);`

`while(y>=x){`

`x++;`

`if(d>0){`

`y--;`

`d=d+4*(x-y)+10;`

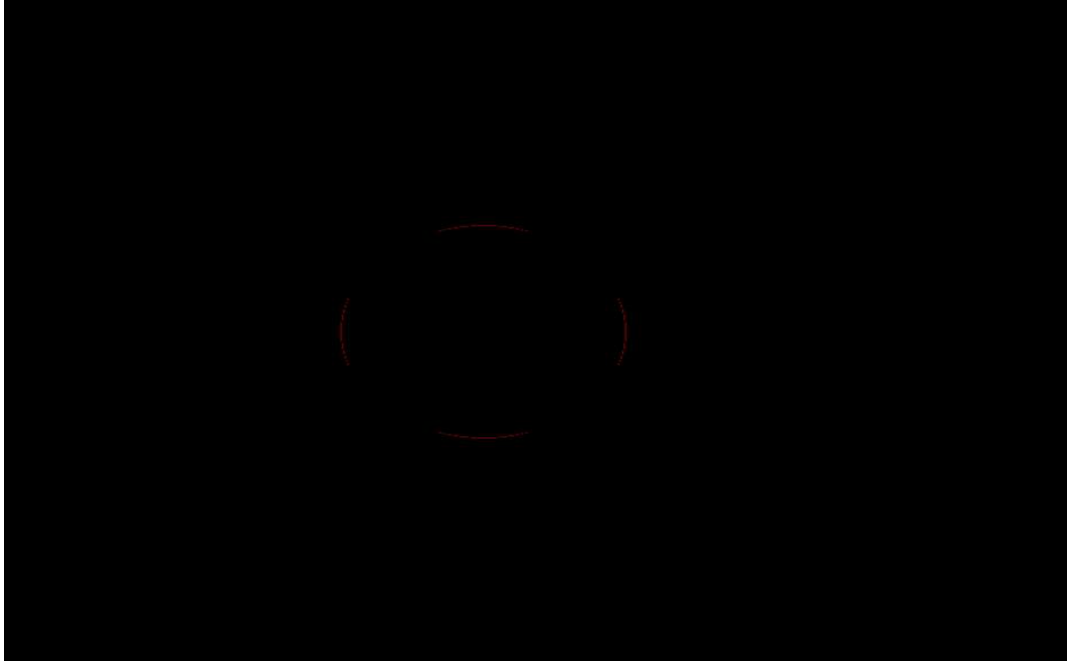
```

    }
    else{
        d=d+4*x+6;
        drawCircle(xc,yc,x,y);
        delay(200);
    }
}
}

int main(){
    int gd=DETECT,gm;
    initgraph(&gd,&gm,NULL);
    initwindow(getmaxwidth(),getmaxheight());
    Circle(getmaxwidth()/2,getmaxheight()/2,100);
    getch();
    closegraph();
    return 0;}

```

OUTPUT: -



Program 6:- Translate a circle along horizontal line in to and fro motion

```
/*  
  
computer graphics practical  
Author:Sachin Parihar  
*/  
  
#include<stdio.h>  
#include<conio.h>  
#include<graphics.h>  
  
void circle_translation(){  
    while(!kbhit()){  
        int j=0;  
        if(j<getmaxwidth()){for(int i=26;i<getmaxwidth();i++){  
            cleardevice();
```

```

        setcolor(RED);
        //Draw circle
        circle(i,getmaxheight()/2,70);
        delay(10);
        j++;
    }
}

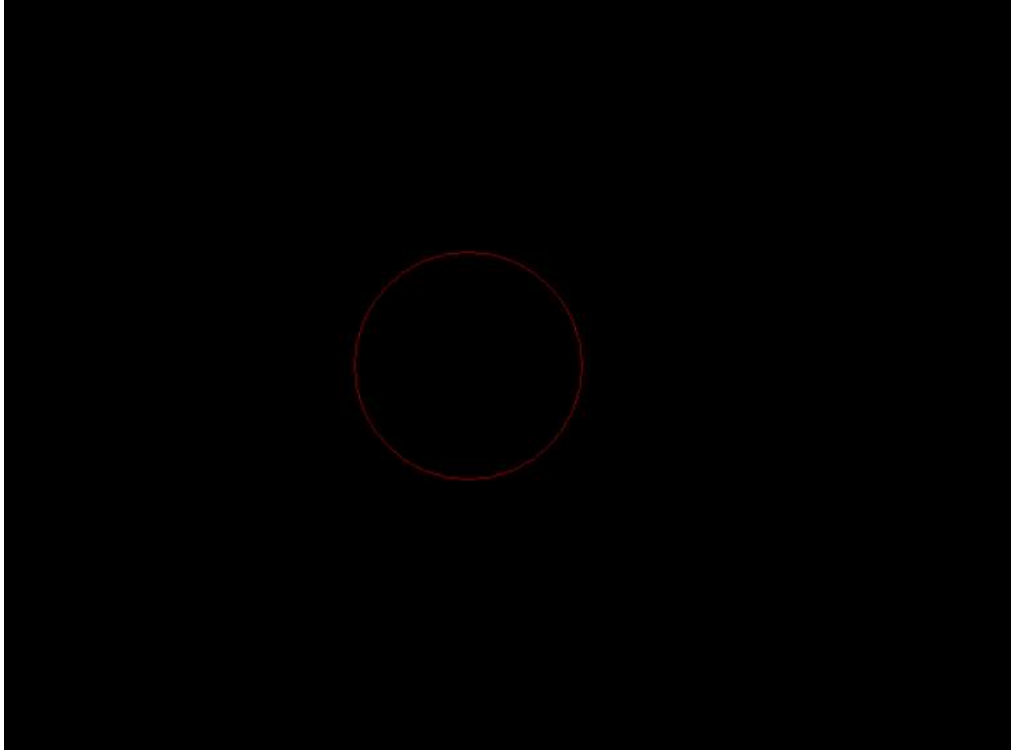
for(int i=getmaxwidth();i>26;i--){
    cleardevice();
    setcolor(RED);
    circle(i,getmaxheight()/2,70);
    delay(10);
}

}

int main(){
    int gd= DETECT,gm;
    initgraph(&gd,&gm,NULL);
    initwindow(getmaxwidth(),getmaxheight());
    circle_translation();
    getch();
    closegraph();
    return 0;
}

```

OUTPUT: -



Program 7:- Rotate a line about it`s center of point

```
/*
```

```
computer graphics practical
```

```
Author:Sachin Parihar
```

```
*/
```

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

```
#include<conio.h>
```

```
#include<graphics.h>
```

```

#include <math.h>

void Line_rotate(int x_a,int y_a,int x_b,int y_b,int theta){
    setcolor(RED);
    line(x_a,y_a,x_b,y_b);
    delay(500);
    cleardevice();

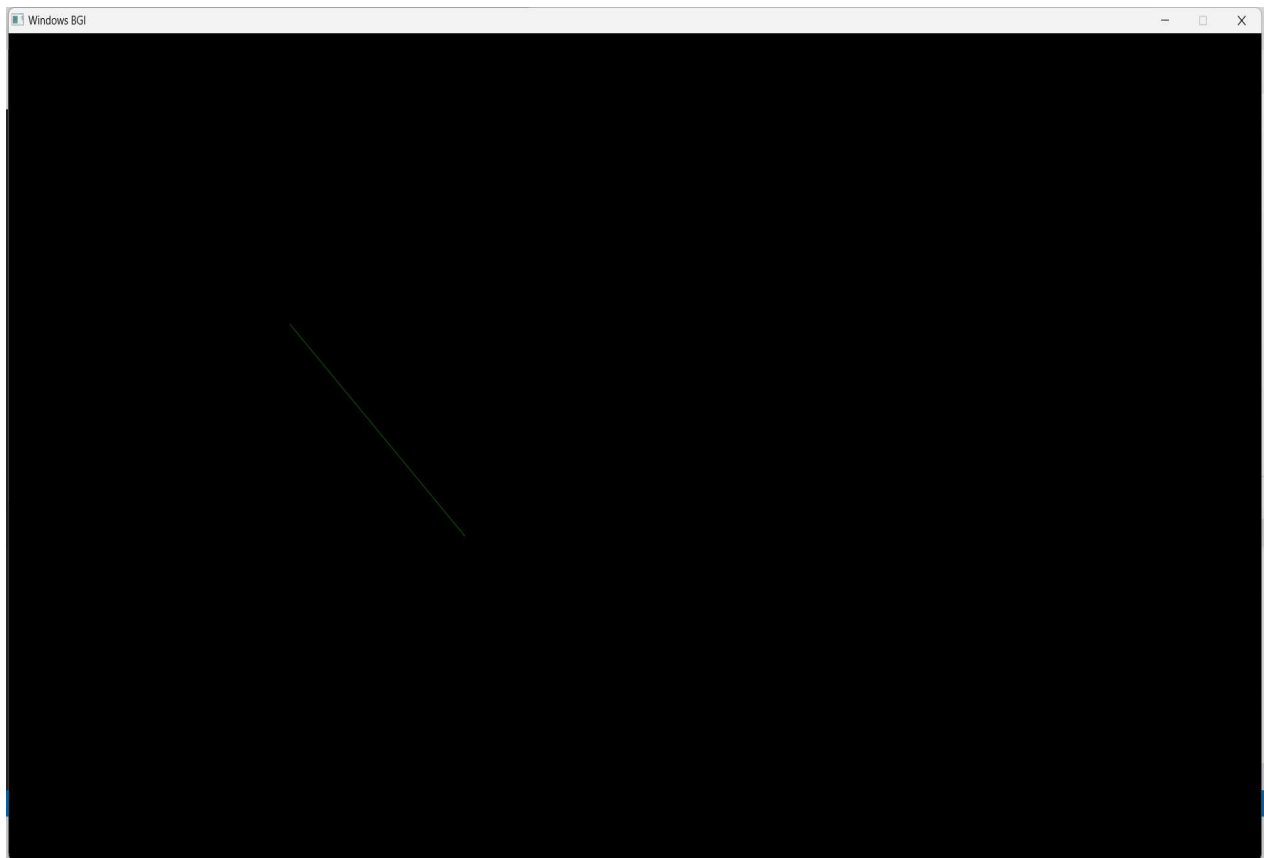
    double Q= theta*M_PI/180.0;
    int x_center = (x_a+x_b)/2;
    int y_center = (y_a+y_b)/2;
    double dx=x_a-x_center, dy=y_a-y_center;
    double newX= dx*cos(Q)-dy*sin(Q);
    double newY= dx*sin(Q)+dy*cos(Q);
    x_a= newX+x_center;
    y_a= newY+y_center;
    dx= x_b-x_center;
    dy=y_b-y_center;
    newX= dx*cos(Q)-dy*sin(Q);
    newY= dx*sin(Q)+dy*cos(Q);
    x_b= newX+x_center;
    y_b= newY+y_center;
    setcolor(GREEN);
    line(x_a,y_a,x_b,y_b);
}

int main(){
    int gd= DETECT,gm;
    initgraph(&gd,&gm,NULL);

```

```
initwindow(getmaxwidth(),getmaxheight());  
Line_rotate(300,400,600,400,45);  
getch();  
closegraph();  
return 0;}
```

OUTPUT:-



Program 8:-scale a rectangle twice/half it's size

```
/*  
computer graphics practical  
Author: Sachin Parihar  
*/  
#include<stdio.h>
```



```

#include<conio.h>

#include<graphics.h>

void Scale_rectangle(int rectx1,int recty1,int rectx2,int recty2,float
sc_factor_x,float scfactor_y){

rectx1 *=sc_factor_x; rectx2
*=sc_factor_x,recty1*=scfactor_y,recty2*=scfactor_y;

setcolor(RED);

rectangle(rectx1,recty1,rectx2,recty2);

}

int main(){

    int gd= DETECT,gm;

    initgraph(&gd,&gm,NULL);

    initwindow(getmaxwidth(),getmaxheight());

    setcolor(GREEN);

rectangle(100,200,300,250);

    Scale_rectangle(100,200,300,250,2,2);

    Scale_rectangle(100,200,300,250,0.5,0.5);

    getch();

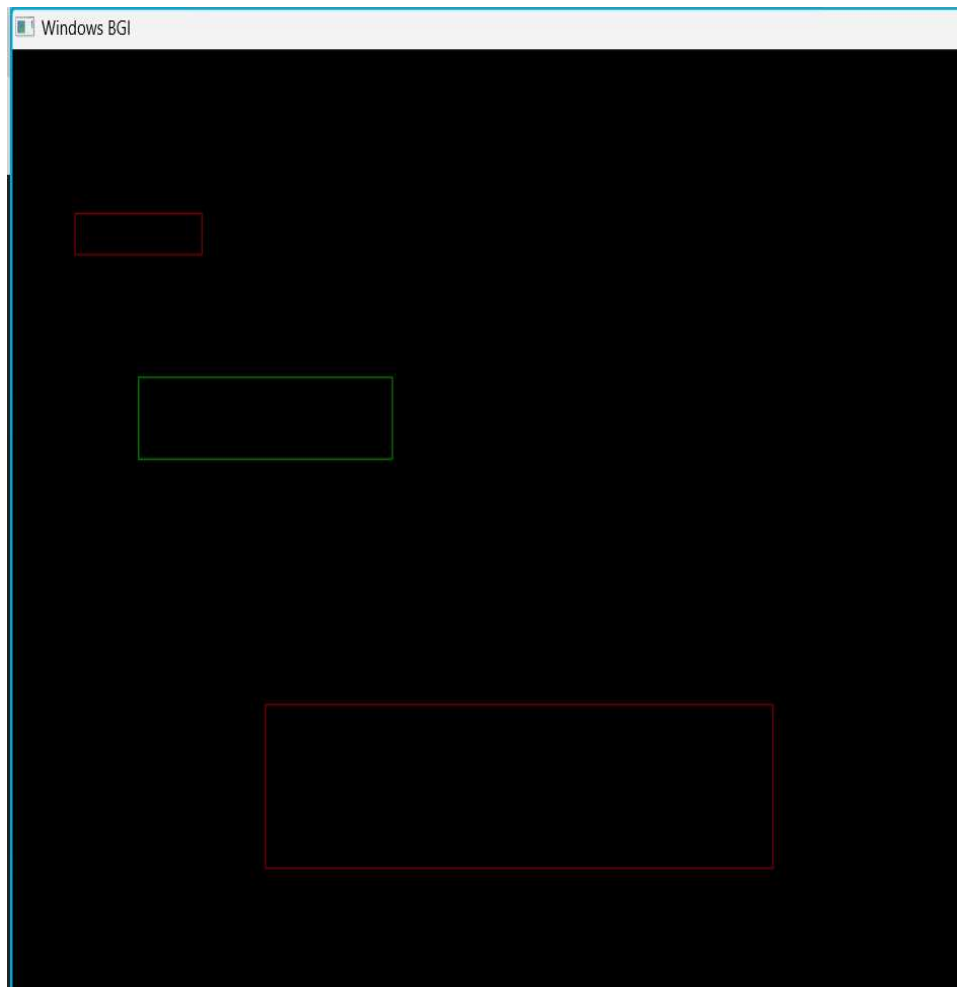
    closegraph();

    return 0;

}

```

OUTPUT: -



Program 9:- implement Cohen Sutherland algorithm for line segment clipping

/*

computer graphics practical

Author: Sachin Parihar

```
*/  
  
#include<stdio.h>  
  
#include<conio.h>  
  
#include<stdbool.h>  
  
#include<graphics.h>  
  
const int inside=0;  
const int left=1;  
const int right=2;  
const int bottom=4;  
const int top=8;  
  
const int x_max=700;  
const int y_max=700;  
const int x_min=200;  
const int y_min=200;  
  
int compute_code(double x,double y){  
    int code=inside;  
    if(x<x_min){  
        code=left;  
    }  
    else if(x>x_max){code=right;}  
    if(y<y_min){code=bottom;}  
    else if(y>y_max){code=top;}  
    return code;  
}
```

```

void cohenSutherlandClip(double x_1,double y_1,double x_2,double y_2){
    int code1= compute_code(x_1,y_1);
    int code2= compute_code(x_2,y_2);
    bool accept=false;
    while(true){
        if((code1==0)&&(code2==0)){
            accept=true;
            break;
        }
        else if(code1 & code2){
            break;
        }
        else{
            int code_out;
            double x,y;
            if(code_out!=0){code_out =code1;}
            else{code_out=code2;}
            if(code_out & top){
                 $x=x_1+(x_2-x_1)*(y_{max}-y_1)/(y_2-y_1);$ 
                y=y_max;
            }
            else if(code_out & bottom){
                 $x=x_1+(x_2-x_1)*(y_{min}-y_1)/(y_2-y_1);$ 
                y=y_min;
            }
            else if (code_out & right){

```

```

        y=y_1+(y_2-y_1)*(x_max-x_1)/(x_2-x_1);
        x=x_max;
    }
    else if(code_out & left){
        y=y_1+(y_2-y_1)*(x_min-x_1)/(x_2-x_1);
        x=x_min;
    }
    if(code_out==code1){
        x_1=x;
        y_1=y;
        code1=compute_code(x_1,y_1);
    }
    else{
        x_2=x;
        y_2=y;
        code2=compute_code(x_2,y_2);
    }
}

if(accept){
    printf("line accepted from %f,%f to %f,%f \n",x_1,y_1,x_2,y_2);
}

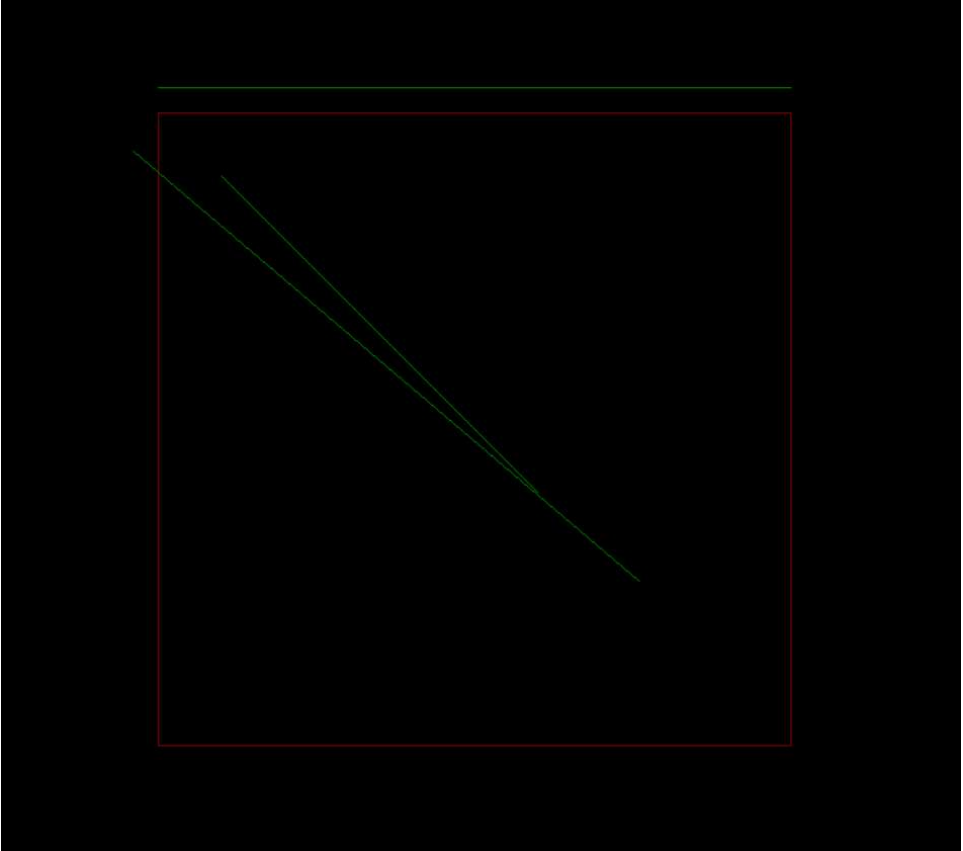
else{printf("line rejected\n");}
}

int main(){

```

```
int gd= DETECT,gm;  
initgraph(&gd,&gm, NULL);  
initwindow(getmaxwidth(),getmaxheight());  
cohenSutherlandClip(50,50,70,70);  
cohenSutherlandClip(250,250,500,500);  
cohenSutherlandClip(200,180,700,180);  
setcolor(RED);  
rectangle(200,200,700,700);  
setcolor(GREEN);  
line(180,230,580,570);  
line(250,250,500,500);  
line(200,180,700,180);  
getch();  
closegraph();  
return 0;  
}
```

OUTPUT:-



PRACTICAL FILE

OF

PYTHON PROGRAMMING LAB



Session-2023-2024

Submitted To: -

Miss. Umesh
(Department Of CSE)

Submitted By: -

Sachin Parihar
Class: - B.Tech CSE 5th Sem
Roll. No. - 210010130099

GJU S&T, Hisar Department of Computer Science and Technology
Guru Jambheshwar University of Science and Technology, Hisar
(Established by State Legislature act 17 of 1995) ("A" Grade NAAC
Accredited)

INDEX

S. No.	TITLE	DATE	SIGNATURE
1.	Install python and explore various IDE like PyCharm		
2.	Assignment to perform number operations		
a)	Program to maximum from a list of numbers		
b)	Program to find GCD of two numbers		
c)	Square root of a number		
d)	Check a number is prime or not		
e)	Print first N prime numbers		
f)	Remove duplicate number from list		
3.	Assignment to perform various operations on a String like Creation, deletion, concatenation		
4.	Program to perform operations on a ListL		
5.	Program to perform operations on a Dictionary.		
6.	Two assignment on Sets to perform operation union, Intersection, Differntiate.		
7.	Two assignment related to searching operation linear search, Binary search.		
8.	Three assignment related to sorting like selection sort, bubble sort, insertion sort.		
9.	Use of dictionary for measuring student marks in five subjects and find maximum , minimum and average marks.		

Program 1. Install Python and explore various IDE like PyCharm.

Python 3 Installation on Windows

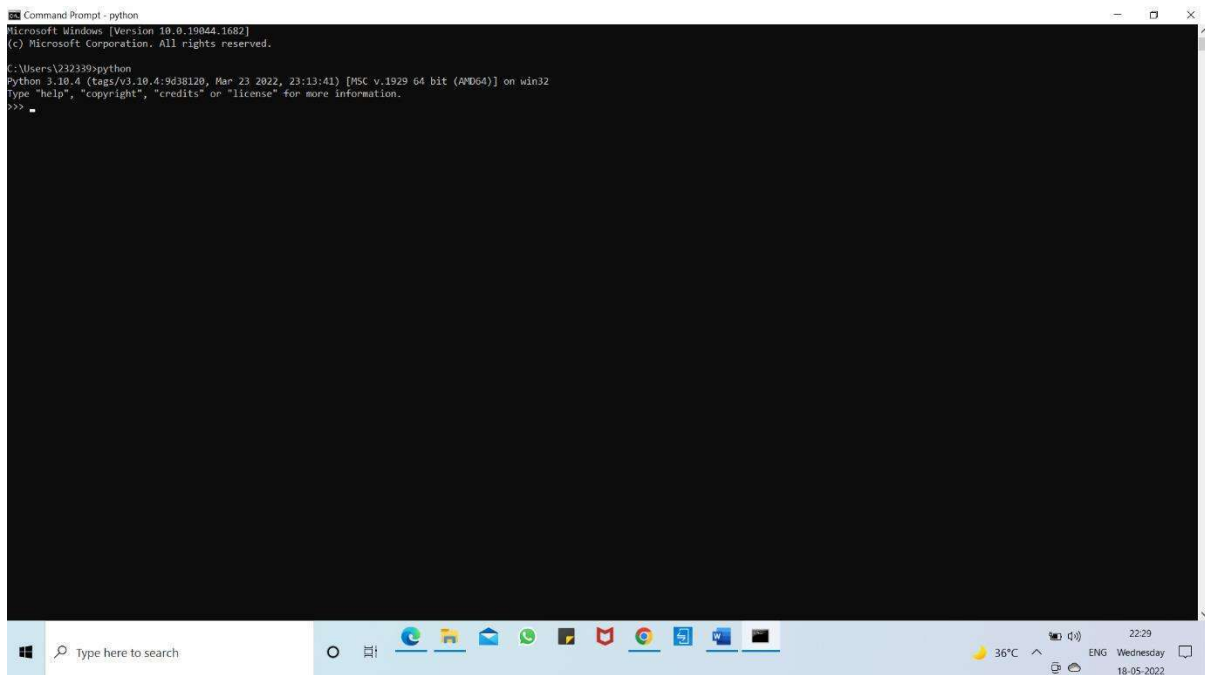
Step 1: Select Version of Python to Install

Step 2: Download Python Executable Installer

Step 3: Run Executable Installer

Step 4: Verify Python Was Installed On Windows

1. Navigate to the directory in which Python was installed on the system. In our case, it is `C:\Users\Username\AppData\Local\Programs\Python\Python37` since we have installed the latest version.
2. Double-click **python.exe**.
3. The output should be similar to what you can see the given picture.

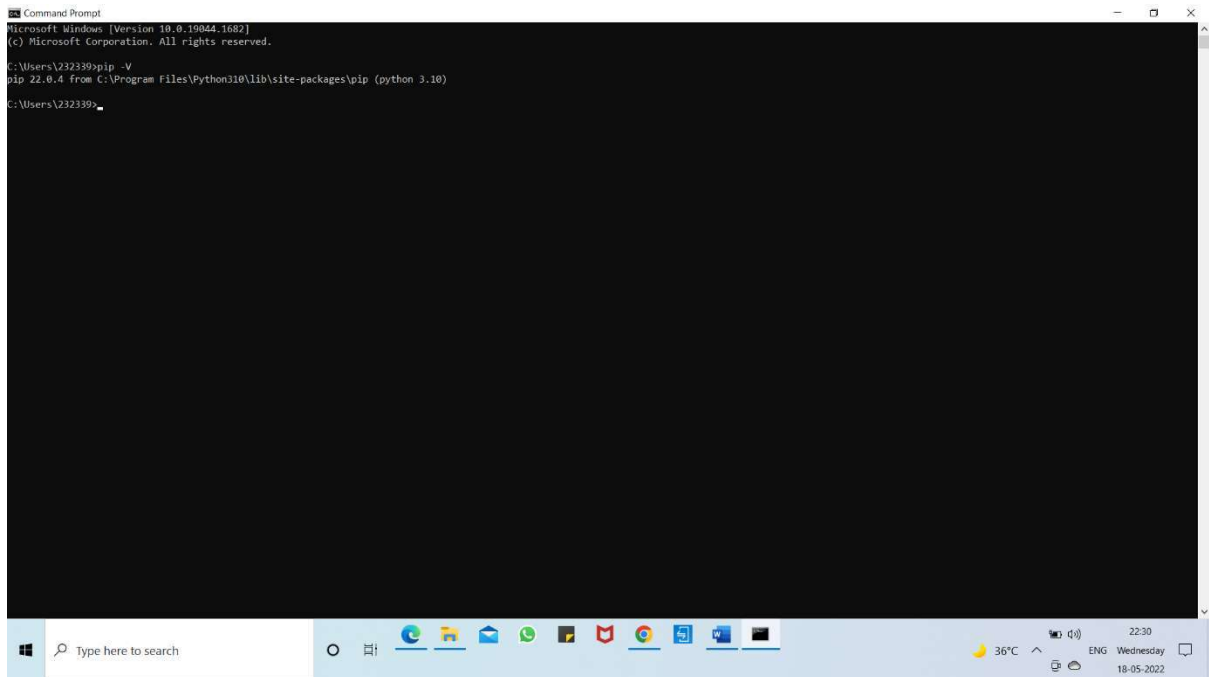


```
Command Prompt - python
Microsoft Windows [Version 10.0.19044.1682]
(c) Microsoft Corporation. All rights reserved.

C:\Users\232339>python
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

Step 5: Verify Pip Was Installed

1. Open the **Start** menu and type "cmd."
2. Select the **Command Prompt** application.
3. Enter `pip -v` in the console. If Pip was installed successfully, you should see the following output in given in picture.



```
Microsoft Windows [Version 10.0.19044.1682]
(c) Microsoft Corporation. All rights reserved.

C:\Users\232339>pip -v
pip 22.0.4 from C:\Program Files\Python310\lib\site-packages\pip (python 3.10)
C:\Users\232339>
```

Step 6: Add Python Path to Environment Variables (Optional)

1. Open the **Start** menu and start the **Run** app.
2. Type `sysdm.cpl` and click **OK**. This opens the **System Properties** window.
3. Navigate to the **Advanced** tab and select **Environment Variables**.
4. Under **System Variables**, find and select the **Path** variable.
5. Click **Edit**.
6. Select the **Variable value** field. Add the path to the `python.exe` file preceded with a **semicolon (;)**. For example, in the image below, we have added `";C:\Python34."`
7. Click **OK** and close all windows.

By setting this up, you can execute Python scripts like this: `Python script.py`
Instead of this: `C:/Python34/Python script.p`

Program 2. Assignment to perform number operation.

a). Maximum from a list of numbers.

Algorithm:-

Step1. Take a function max(list) then assign maximum value of list

max = list[0]

Step2. Initialise for loop i.e.,

for i = 1 , for which i < n

now apply if condition i.e.,

if list[i] > max

max = list[i]

Step3. Now return the maximum value .

Program:-

```
# Python program to find largest
```

```
# number in a list
```

```
# list of numbers
```

```
list1 = [145, 459, 568, 3456, 94556, 45563]
```

```
# sorting the list
```

```
list1.sort()
```

```
# printing the last element
```

```
print("Largest element is:", list1[-1])
```

Output:-

C:\Users\232339\PycharmProjects\Myproject\venv\Scripts\python.exe

C:/Users/232339/PycharmProjects/Myproject/prog.py

Largest element is: 94556

b). Program to find GCD of two numbers.

Algorithm:-

1. Let a, b be the two numbers.
2. $a \bmod b = R$.
3. Let $a = b$ and $b = R$.
4. Repeat Steps 2 and 3 until $a \bmod b$ is greater than 0.
5. $GCD = b$.
6. Finish.

Program:-

```
# Python code to demonstrate the working of
gcd()

# importing "math" for mathematical operations
import math

# prints 12
print("The gcd of 60 and 48 is : ", end="")

print(math.gcd(80, 98))
```

Output:-

C:\Users\232339\PycharmProjects\Myproject\venv\Scripts\python.exe

C:/Users/232339/PycharmProjects/Myproject/prog.py

The gcd of 60 and 48 is : 2

c). Program to find square root of a number

ALGORITHM :-

Step1. Import the math module using import math at the beginning of the program.

Step2. Find out the square root of the number using math.sqrt(number) method.

Step3. Note that it throws an error if a negative number is passed.

Program:-

```
import math
number = int(input("enter a number:"))
sqrt = math.pow(number, 0.5)
print("square root: ", sqrt)
```

Output:-

```
C:\Users\232339\PycharmProjects\Myproject\venv\Scripts\python.exe
```

```
C:/Users/232339/PycharmProjects/Myproject/prog.py
```

```
enter a number:169
```

```
square root: 13.0
```

```
enter a number:458
```

```
square root: 21.400934559032695
```

```
enter a number:90134
```

```
square root: 300.2232502655316
```

d).To check number is prime or not.

ALGORITHM:-

Step 1: Start

Step 2: Initialize variables num,flag=1, j=2

Step 3: Read num from user

Step 4: If num<=1 // Any number less than 1 is not a prime number
Display "num is not a prime number" Goto step 7

Step 5: Repeat the steps until $j < [(n/2)+1]$

5.1 If remainder of number divide j equals to 0,

Set flag=0

Goto step 6

5.2 $j=j+1$

Step 6: If flag==0,

Display num+" is not prime number"

Else

Display num+" n is prime number"

Step 7: Stop

PROGRAM:-

To check a number is prime or not.

Program to check if a number is prime or not

```
num = 407
```

```
# To take input from the user
```

```
#num = int(input("Enter a number: "))
```

```
# prime numbers are greater than 1
```

```
if num > 1:
```

```
    # check for factors
```

```
    for i in range(2,num):
```

```
        if (num % i) == 0:
```

```
            print(num,"is not a prime number")
```

```
            print(i,"times",num//i,"is",num)
```

```
            break
```

```
    else:
```

```
        print(num,"is a prime number")
```

```
# if input number is less than
```

```
# or equal to 1, it is not prime
```

```
else:
```

```
    print(num,"is not a prime number")
```

Output:-

893 is not a prime number

19 times 47 is 893

e). PROGRAM:- Print first N prime numbers.

ALGORITHM:-

Step 1: Loop through all the elements in the given range

Step 2: Check for each number if it has any factor between 1 and itself.

Step 3: If yes, then the number is not prime, and it will move to the next number.

Step 4: If no, it is the prime number, and the program will print it and check for the next number.

Step 5: The loop will break when it is reached to the upper value.

PROGRAM:-

First, we will take the input:

```
lower_value = int(input("Please, Enter the Lowest Range Value: "))
```

```
upper_value = int(input("Please, Enter the Upper Range Value: "))
```

```
print("The Prime Numbers in the range are: ")
```

```
for number in range(lower_value, upper_value + 1):
```

```
    if number > 1:
```

```
        for i in range(2, number):
```

```
            if (number % i) == 0:
```

```
                break
```

```
        else:
```

```
            print(number)
```

Output:-

Please, Enter the Lowest Range Value: 1

Please, Enter the Upper Range Value: 45

The Prime Numbers in the range are:

2

3

5

7

11

13

17

19

23

29

31

37

41

43

Process finished with exit code 0

f). Program To remove duplicate numbers from list.

ALGORITHM:-

1. Method 1: Naïve Method.
2. Method 2: Using a list comprehensive.
3. Method 3: Using set()
4. Method 4: Using list comprehensive + enumerate()
5. Method 5: Using collections. OrderedDict. fromkeys()

Program:- # Python 3 code to demonstrate

removing duplicated from list

using naive methods

initializing list

```
test_list = [17, 23, 15, 6, 23, 15, 6, 17]
```

```
print ("The original list is : " + str(test_list))
```

using naive method

to remove duplicated

from list

```
res = []
```

```
for i in test_list:
```

```
    if i not in res:
```

```
        res.append(i)
```

printing list after removal

```
print ("The list after removing duplicates : " + str(res))
```


Output:-

The original list is : [17, 23, 15, 6, 23, 15, 6, 17]

The list after removing duplicates : [17, 23, 15, 6]

Process finished with exit code 0

g). PROGRAM To print Fibonacci series.

ALGORITHM:-

Initialize variables a,b to 1

Initialize for loop in range[1,n) # n exclusive

Compute next number in series; total = a+b

Store previous value in b

Store total in a

Program to display the Fibonacci sequence up to n-th term

```
nterms = int(input("How many terms? "))
```

```
# first two terms
```

```
n1, n2 = 0, 1
```

```
count = 0
```

```
# check if the number of terms is valid
```

```
if nterms <= 0:
```

```
    print("Please enter a positive integer")
```

```
# if there is only one term, return n1
```

```
elif nterms == 1:
```

```
    print("Fibonacci sequence upto",nterms,":")
```

```
    print(n1)
```

```
# generate fibonacci sequence
```

```
else:
```

```
    print("Fibonacci sequence:")
```

```
    while count < nterms:
```

```
        print(n1)
```

```
        nth = n1 + n2
```

```
        # update values
```

```
        n1 = n2
```

```
        n2 = nth
```

```
        count += 1
```

Output:-

How many terms? 12

Fibonacci sequence:

0

1

1

2

3

5

8

13

21

34

55

89

Process finished with exit code 0

Program 2. To perform various operations on Strings like creation, deletion, concatenation.

ALGORITHM:-

Strings can be created by enclosing characters inside a single quote or double-quotes.

Even triple quotes can be used in Python but generally used to represent multiline strings and docstrings

Program:-

```
# defining strings in Python
```

```
# all of the following are equivalent
```

```
my_string = 'Hello'
```

```
print(my_string)
```

```
my_string = "Hello"
```

```
print(my_string)
```

```
my_string = """Hello"""
```

```
print(my_string)
```

```
my_string= '''world'''
```

```
print(my_string)
```

```
# triple quotes string can extend multiple lines
```

```
my_string = """Hello, welcome to
```

```
the world of Python"""
```

```
print(my_string)
```

Output:-

C:\Users\232339\PycharmProjects\Myproject\venv\Scripts\python.exe

C:/Users/232339/PycharmProjects/Myproject/sec.py

Hello

Hello

Hello

world

Hello, welcome to

the world of Python

Process finished with exit code 0

Program:- For Deletion

Algorithm:-

You can remove a character from a Python string using `replace()` or `translate()`.

Both these methods replace a character or string with a given value.

If an empty string is specified, the character or string you select is removed from the string without a replacement

Deletion:-

```
input_str = "SurajPartap"
```

```
# Printing original string
```

```
print ("Original string: " + input_str)
```

```
result_str = ""
```

```
for i in range(0, len(input_str)):
```

```
    if i != 4:
```

```
        result_str = result_str + input_str[i]
```

```
# Printing string after removal
```

```
print ("String after removal of i'th character : " + result_str)
```

Output:-

Original string: SurajPartap

String after removal of i'th character : SuraPartap

PROGRAM:- FOR CONCATENATION

ALGORITHM:-

Given two Strings, the task is to concatenate the two Strings without using another string and without using concat function

Concatenate :- Concatenation means combining two strings together.

Example :- Input: str1 = "letsfind", str2 = "course"

PROGRAM:-

```
# Python program to demonstrate
```

```
# string concatenation
```

```
# Defining strings
```

```
var1 = "Raj "
```

```
var2 = "Kumar"
```

```
# + Operator is used to combine strings
```

```
var3 = var1 + var2
```

```
print(var3)
```

```
var4= "very"
```

```
var5= "good"
```

```
var6= var4 + var5
```

```
print(var6)
```

Output:-

Raj Kumar

verygood

Process finished with exit code 0

PROGRAM :-4 Create a list L[10,20,30].Write programs to perform following operations

Insert new numbers to ListL

Delete numbers from listL

Sum all numbers in listL

Delete the ListL.

Algorithm

##insert new number

Syntax: list_name.insert(index, element)

Parameters:

Returns: This method does not return any value but it inserts the given element at the given index.

Error: ...

Note:

Delete

1. remove() remove() deletes the first instance of a value in a list. ...

2. del. del can be used to delete a single index of a list, a slice of a list, or the complete list.

...

3. pop() The pop method removes an element at a given index and returns its value

#Sum all number in ListL

```
? def sum_of_list(l): total = 0. for val in l: total = total + val.  
return total. my_list =
```

```
[1,3,5,2,4] ...
```

```
? def sum_of_list(l,n): if n == 0: return l[n]; return l[n] +  
sum_of_list(l,n-1) my_list =
```

```
[1,3,5,2,4] ...
```

```
? my_list = [1,3,5,2,4] print "The sum of my_list is",  
sum(my_list) Run.
```

#Delete the list

1. Syntax: list_name.remove(obj)

2. Parameters:

3. Returns: The method does not return any value but removes the given object from the

list.

4. Exception: ...

5. Note:

Progran:-

```
ListL= [10,20,30]
```

#insert element in list

```
ListL.insert(50,60)
```

```
print(ListL)
```

```
#delete from list
```

```
ListL.remove(30)
```

```
print(ListL)
```

```
# Python code to explain working on sum() method
```

```
# Declare list of numbers
```

```
numlistL = [10,20,30,40,33,17,11,15]
```

```
numsum = sum(numlistL)
```

```
print('Sum of List: ',numsum)
```

```
# Example with start
```

```
numsum = sum(numlistL, 5)
```

```
print('Sum of List: ',numsum)
```

```
#to find sum of prime number
```

```
def findPrime(listL):
```

```
    s = 0
```

```
    for num in listL:
```

```
        i = 2
```

```
        p = 1
```

```
        while i <= num / 2:
```

```
if num % i == 0:
    p = 0
    break
    i = i + 1
if p == 1:
    s = s + num
return s

listL = [10, 11, 12, 13, 14, 15]
s = findPrime(listL)
print("Sum of all prime numbers:", s)

#delete the list

listL = [10,20,30,0, 4, 1]
print('listL before clear:', listL)

# Clearing list
listL.clear()
print('listL after clear:', listL)
```

Output:-

[10, 20, 30, 60]

[10, 20, 60]

Sum of List: 176

Sum of List: 181

Sum of all prime numbers: 24

listL before clear: [10, 20, 30, 0, 4, 1]

listL after clear: []

Program:- 5. Create a Dictionary D={'Name': 'Allen', 'Age':27 , 5:123456} program to perform various operations.

- a.Insert new entry in D
- b.Delete an entry fromD
- c. Check whether a key present in D.
- d. Clear dictionaryD.

Algorithm

#Insert

Appending element(s) to a dictionary

To append an element to an existing dictionary, you have to use the dictionary name followed

by square brackets with the key name and assign a value to it.

#Delete

del keyword can be used to inplace delete the key that is present in the dictionary. One

drawback that can be thought of using this is that it raises an exception if the key is not found

and hence non-existence of key has to be handled. pop() can be used to delete a key and its

value inplace

#to key present in D

Checking if key exists using the get() method

The `get()` method is a dictionary method that returns the value of the associated key. If

the key is not present it returns either a default value (if passed) or it returns `None`.

Using this method we can pass a key and check if a key exists in the python dictionary.

#Clear Dictionary

Syntax: `dict.clear()`

Parameters: The `clear()` method doesn't take any parameters.

Returns: The `clear()` method doesn't return any value.

Examples: Input : `d = {1: "geeks", 2: "for"}` `d.clear()` Output :
`d = {}`

Error: ...

Output: `text = {}`

Program:-

```
D = {'Name': 'Allen', 'Age': 27, 5: 123456}
```

```
print(D)
```

```
#insert new entry in D
```

```
D['class'] = 'MCA'
```

```
print(D)
```

```
#delete an entry From D
```

```
del D['Age']
```

```
print(D)
#if 'Name' in D:
print("this will execute")
if "nonexistent key" in D:
    print("this will not")
#to update the value of the key
D.update({'Name':'Ritu'})
print(D)
# to clear dictionary D
D.clear()
```

Output:-

C:\Users\232339\PycharmProjects\Myproject\venv\Scripts\python.exe

C:/Users/232339/PycharmProjects/Myproject/sec.py

{'Name': 'Allen', 'Age': 27, 5: 123456}

{'Name': 'Allen', 'Age': 27, 5: 123456, 'class': 'MCA'}

{'Name': 'Allen', 5: 123456, 'class': 'MCA'}

this will execute

{'Name': 'Ritu', 5: 123456, 'class': 'MCA'}

Process finished with exit code 0

Program:- 6. Two assignment on sets to perform various operation like union, intersection, differentiate.

Algorithm

Union

To implement the Union-Find in Python, we use the concept of trees. The

tree's root can act as a representative, and each node will hold the reference

to its parent node.

The Union-Find algorithm will traverse the parent nodes to reach the root and

combine two trees by attaching their roots.

Intersection :-

Python intersection() function return a new set with an element that is

common to all set

The intersection of two given sets is the largest set which contains all the

elements that are common to both sets

Differentiate:-

The difference() method in Python returns the difference between two given sets.

Lets say we have two sets X and Y, the difference between X and Y is denoted by $X - Y$

and it contains the elements that are in Set X but in Set Y.

Program:-

```
# Program to perform different set operations
```

```
# as we do in mathematics
```

```
# sets are define
```

```
X = {0, 2, 4, 6, 8,9};
```

```
Y = {1, 2, 3, 4, 5,7};
```

```
# union
```

```
print("Union :", X | Y)
```

```
# intersection
```

```
print("Intersection :", X & Y)
```

```
# difference
```

```
print("Difference :", X - Y)
```

```
# symmetric difference
```

```
print("Symmetric difference :", X ^ Y)
```

Output:-

C:\Users\232339\PycharmProjects\Myproject\venv\Scripts\python.exe

C:/Users/232339/PycharmProjects/Myproject/sec.py

Union : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

Intersection : {2, 4}

Difference : {0, 8, 6, 9}

Symmetric difference : {0, 1, 3, 5, 6, 7, 8, 9}

Process finished with exit code 0

PROGRAM:-7. Two assignment related to searching operation like linear

search, binary search.

Algorithm :-

#Linear search

LinearSearch(array, key)

for each element in the array

if element == value

return its index

#Binary search

do until the pointers low and high are equal.

mid = (low + high)/2

if (k == arr[mid])

return mid

else if (k > arr[mid]) // k is on right side of mid

low = mid + 1

else // k is on left side of mid

high = mid - 1

Program:-

#linear search

```
def LinearSearch(array, n, k):
```

```
    for j in range(0, n):
```

```
        if (array[j] == k):
```

```
            return j
```

```
    return -1
```

```
array = [11, 33, 85, 97, 99]
```

```
k = 85
```

```
n = len(array)
```

```
result = LinearSearch(array, n, k)
```

```
if (result == -1):
```

```
    print("Element not found")
```

```
else:
```

```
    print("Element found at index: ", result)
```

#binary search

```
def binarySearch(arr, k, low, high):
```

```
    while low <= high:
```

```
        mid = low + (high - low) // 2
```

```
        if arr[mid] == k:
```

```
            return mid
```



```
elif arr[mid] < k:
```

```
low = mid + 1
```

```
else:
```

```
high = mid - 1
```

```
return -1
```

```
arr = [1, 3, 5, 7, 9]
```

```
k = 97
```

```
result = binarySearch(arr, k, 0, len(arr) - 1)
```

```
if result != -1:
```

```
print("Element is present at index: " + str(result))
```

```
else:
```

```
print("Not found")
```

Output:-

Element found at index: 2

Element is present at index: 3

PROGRAM:-8. Three assignment related to sorting like selection sort, bubble, sort, insertion sort.

Algorithm :-

```
#bubble sort
```

```
for i in range len(L):
```

```
for j in range (0,len(L)-1):
```

```
if (L[j]>L[j+1]):
```

```
temp=L[j]
```

```
L[j]=L[j+1]
```

```
L[j+1]=temp
```

Program:-

```
L=[]
```

```
n=int(input("Enter number of elements"))
```

```
print("Enter the elements of lists:")
```

```
for item in range(n):
```

```
    a=int(input(" "))
```

```
    L.append(a)
```

```
print("List is: ",L)
```

```
for i in range(len(L)):
```

```
    for j in range(0,(len(L)-1)-i):
```

```
        if (L[j]>L[j+1]):
```

```
            temp=L[j]
```

```
            L[j]=L[j+1]
```

```
            L[j+1]=temp
```

```
print("sorted List is:",L)
```

Output:-

Enter number of elements 8

Enter the elements of lists:

12

28

33

98

55

66

45

87

lList is: [12, 28, 33, 98, 55, 66, 45, 87]

sorted List is: [12, 28, 33, 45, 55, 66, 87, 98]

#Selection sort

Algorithm:-

1. Create a function Selection_Sort that takes an array as an argument
2. Create a loop with a loop variable i that counts from 0 to the length of the array –
1
3. Declare smallest with the initial value i
4. Create an inner loop with a loop variable j that counts from i + 1 up to the length
of the array – 1.
5. if the elements at index j are smaller than the element at index smallest, then set
smallest equal to j
6. swap the elements at indexes i and smallest
7. Print the sorted list

Program:-

```
def Selection_Sort(array):  
    for i in range(0, len(array) - 1):  
        smallest = i  
        for j in range(i + 1, len(array)):
```

```
if array[j] < array[smallest]:  
    smallest = j  
    array[i], array[smallest] = array[smallest], array[i]  
array = input('Enter the list of numbers: ').split()  
array = [int(x) for x in array]  
Selection_Sort(array)  
print('List after sorting is : ', end="")  
print(array)
```

OUTPUT:-

Enter the list of numbers: 4 6 8 11 56 89 15 67

List after sorting is : [4, 6, 8, 11, 15, 56, 67, 89]

PROGRAM:-9. Demonstrate the use of dictionary for measuring student marks in five subject and you have to find the student having maximum and minimum average marks.

ALGORITHM:-

In Python to find the minimum and maximum value in a dictionary, we can use the built-in min() and max() function.

In Python the max() function is used to find the maximum values in a given dictionary.

While in the case of min() function is used to find the minimum values in a dictionary

Average

You can do this by iterating over the dictionary and filtering out zero values

first.

Then take the sum of the filtered values.

Finally, divide by the number of these filtered values

PROGRAM:-

```
D={'PYTHON':198,'DBMS':167,'DSA':289,'CSA':278,'ARTIFICIAL  
INTELLIGENCE':190}
```

```
print(D)
```

```
#maximum marks
```

```
keymax=max(zip(D.values(),D.keys()))[1]
```



```
print(keymax)

#minimum marks

keymax=min(zip(D.values(),D.keys()))[1]

print(keymax)

#average

# printing original dictionary
print("The original dictionary is : " + str(D))

# loop to sum all values

res = 0

for val in D.values():

    res += val

# using len() to get total keys for AVERAGE computation

res = res / len(D)

# printing result

print("The AVERAGE IS : " + str(res))
```

OUTPUT:-

C:\Users\232339\PycharmProjects\Myproject\venv\Scripts\python.exe

C:/Users/232339/PycharmProjects/Myproject/sec.py

{'PYTHON': 198, 'DBMS': 167, 'DSA': 289, 'CSA': 278, 'ARTIFICIAL INTELLIGENCE': 190}

DSA

DBMS

The original dictionary is : {'PYTHON': 198, 'DBMS': 167, 'DSA': 289, 'CSA': 278, 'ARTIFICIAL INTELLIGENCE': 190}

The AVERAGE IS : 224.4

Process finished with exit code 0