**Practical No: 14**

**Aim :** Write a C program to implement the midpoint subdivision algorithm.

**Performed By :**

**Class :** BCA-III SEM-V

**Date :** 10/09/2024

**Code :**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

#include<dos.h>

#include<stdlib.h>

typedef struct coordinates

{

int x;

int y;

char code[4];

}pt;

pt p1,p2,ptemp;

void drawwindow();

void drawline(pt p1,pt p2, int c1);

pt setcode(pt p);

int visibility(pt p1,pt p2);

pt resetendpt(pt p1,pt p2);

void midsub(pt p1,pt p2);

int main(){

int gd=DETECT,gm,v;

detectgraph(&gd,&gm);

initgraph(&gd,&gm,"c:\\tc\\bgi");

cleardevice();

printf("Enter the endpoint 1\t");

scanf("%d %d",&p1.x,&p1.y);

printf("\nEnter the endpoint 2\t");

scanf("%d %d",&p2.x,&p2.y);

cleardevice();

drawwindow();

drawline(p1,p2,15);

getch();

cleardevice();

drawwindow();

midsub(p1,p2);

getch();

closegraph();

}

void midsub(pt p1,pt p2){

pt mid;

int v;

p1=setcode(p1);

p2=setcode(p2);

v=visibility(p1,p2);

delay(100);

switch(v)

{

case 0:

{

drawline(p1,p2,15);

break;

}

case 1:

{

break;

}

case 2:

{

mid.x=p1.x+(p2.x-p1.x)/2;

mid.y=p1.y+(p2.y-p1.y)/2;

midsub(p1,mid);

mid.x=mid.x+1;

mid.y=mid.y+1;

midsub(mid,p2);

break;

}

}

}

void drawwindow(){

setcolor(RED);

line(150,100,450,100);

line(450,100,450,400);

line(450,400,150,400);

line(150,400,150,100);

}

void drawline(pt p1,pt p2,int c1)

{

setcolor(c1);

line(p1.x,p1.y,p2.x,p2.y);

}

pt setcode(pt p)

{

pt ptemp;

if(p.y>400)

ptemp.code[0]='1';

else

ptemp.code[0]='0';

if(p.y<100) ptemp.code[1]='1'; else ptemp.code[1]='0'; if(p.x>450)

ptemp.code[2]='1';

else

ptemp.code[2]='0';

if(p.x<150)

ptemp.code[3]='1';

else

ptemp.code[3]='0';

ptemp.x=p.x;

ptemp.y=p.y;

return(ptemp);

}

int visibility(pt p1,pt p2)

{

int i,flag=0;

for(i=0;i<4;i++)

{

if((p1.code[i]!='0')||(p2.code[i]!='0'))

flag=1;

}

if(flag==0)

return(0);

for(i=0;i<4;i++)

{

if((p1.code[i]==p2.code[i])&&(p1.code[i]=='1'))

flag=0;

}

if(flag==0)

return(1);

return(2);

}

**Output :**

