[08:07, 26/03/2024] Sejal Pisal: AIM: Study and enlist the basic functions used for graphics in C/C++ language. Give an example for each of them.

ETPROGRAM

```
#include <graphics.h>

#include<conio.h>

main()

int

}

0

0

gd=DETECT.gm.left=100.top=100,right 200,bottom=200,

x=300,y=150,radius=50;

initgraph(&gd,&gm,"c:\\TURBOC3\\BGI");

rectangle(left,top,right,bottom);

circle(x,y,radius);

bar(left+300,top, right +300,bottom);
```

```
line(left-10,top+150,left+410,top+150);

ellipse(x,y+200,0,360,100,50);

outtextxy(left+100,top+325,"My First C Graphics program");

getch();

closegraph();

return 0;

}
```

Output

en
[08:08, 26/03/2024] Sejal Pisal: AIM: Draw a co-ordinate axis at the center of the screen.

PROGRAM

```
#include <graphics.h>

#include <conio.h>

main()

{

int gd=DETECT,gm;
int midx,midy;

initgraph(&gd,&gm, "C:\\TURBOC3\\BGI");

midx=getmaxx()/2;

midy=getmaxy()/2;

line(1,midy,640,midy);

line(midx,1,midx,6-40);

getch();

closegraph();

return 0;

}
```

[08:14, 26/03/2024] Sejal Pisal: J

PRACTICAL NO: 2A

AIM: Divide your screen into four region, draw circle, rectangle, ellipse and half ellipse in each region with appropriate message.

PROGRAM

```
#include<stdio.h>

#include<conio.h>
```

```
#include<graphics.h>

main()

{

int gd=DETECT,gm;

int midx,midy;

initgraph(&gd,&gm, "c:\\TURBOC3\\bgi");

midx=getmaxx()/2;

midy=getmaxy()/2;

line(1,midy, 840, midy);

line(midx, 1, midx,940);

circle (150,130,50);

outtextxy(130,200, "CIRCLE");

rectangle(400,90,500,170);

outtextxy(420,200, "RECTANGLE");

arc(150,350,0,180,50);
outtextxy(140,380,"ARC");
ellipse(450,320,0,360,50,40);

outtextxy(425,375,"ELLIPSE");

getch();

closegraph();

return 0;

}
```

Output

[08:14, 26/03/2024] Sejal Pisal: AIM: Draw a simple hut on the screen.

PROGRAM

```
#include<graphics.h>

#include<conio.h>

int main() {

int gd DETECT,gm;

initgraph(&gd, &gm, "c:\\TURBOC3\\BGI");

/* Draw Hut */

setcolor(WHITE);
rectangle(150,180,250,300);

rectangle (250,180,420,300);

rectangle(180,250,220,300);

line(200,100,150,180);

line(200,100,250,180);

line(200,100,370,100);

line(370,100,420,180);

getch();

closegraph();

return 0;

}
```

[08:15, 26/03/2024] Sejal Pisal: AIM: Draw the following basic shapes in the center of the screen:

i. Circle

ii. Rectangle

iii. Square

iv. Concentric Circles

v. Ellipse

vi. Line

PROGRAM

i. Circle

```
#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main() {

int gd = DETECT,gm;

int x,y,radius=80;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

x = getmaxx()/2;

y = getmaxy()/2;

outtextxy(160,50, "BASIC SHAPE AT THE CENTER OF SCREEN-CIRCLE");

circle(x, y, radius);
```

```
getch();

closegraph();

return 0;
```
[08:15, 26/03/2024] Sejal Pisal: of

ii. Rectangle

```
#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main() {

int gd DETECT.gm;

int x,y;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

outtextxy(160,50, "BASIC SHAPE AT THE CENTER OF

SCREEN- RECTANGLE");

rectangle(170,420,500,170);

getch();
closegraph();

return 0;

}
```

[08:16, 26/03/2024] Sejal Pisal: iii. Square

```
#include<stdio.h>

#include <graphics.h>

#include<conio.h>

int main() {

int gd = DETECT,gm;

int x,y;

initgraph(&gd, &gım, "C:\\TURBOC3\\BGI");

outtextxy (160,50, "BASIC SHAPE AT THE CENTER OF

SCREEN-SQUARE");

rectangle(250,180,380,340);

getch();

closegraph();
return 0;

}
```

[08:16, 26/03/2024] Sejal Pisal: iv. Concentric Circles

```
#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main() {

int gd = DETECT,gm,color=1;

int x,y,i;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

x = getmaxx()/2;

y = getmaxy()/2;

outtextxy(160,20, "BASIC SHAPE AT THE CENTER OF

SCREEN-CIRCLE");

for(i=20;i<=200;i+=20){

setcolor(color++);
circle(x,y,i); }

getch();

closegraph();

return 0;

}
```

[08:16, 26/03/2024] Sejal Pisal: v. Ellipse

```
#include<stdio.h>

#include <graphics.h>

#include<conio.h>

int main() {

int gd DETECT.gm;

int x,y;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

x = getmaxx()/2;

y = getmaxy()/2;

outtextxy(160,50, "BASIC SHAPE AT THE CENTER OF
SCREEN-ellipse");

ellipse(x, y, 0, 360, 120, 60);

getch();

closegraph();

return 0;

}
```
[08:17, 26/03/2024] Sejal Pisal: vi. Line

```
#include<stdio.h>

#include<graphics.h>

#include<conio.h>

int main() {

int gd DETECT.gm;

int x,y;

initgraph(&gd. & gm, "C:\\TURBOC3\\BGI");

x= getmaxx()/2;

y=getmaxy()/2;
outtextxy (160,50, "BASIC SHAPE AT THE CENTER OF

SCREEN-LINE");

line(100,250,500,250);

getch();

closegraph();

return 0;

}
```

[08:18, 26/03/2024] Sejal Pisal: AIM: Develop the program for DDA Line drawing algorithm.

PROGRAM

```
#include <graphics.h>

#include <conio.h>

#include <math.h>

void main()

{

float x,y,x1,y1,x2,y2,dx,dy,step;

int i,gd=DETECT.gm;
```

```
initgraph(&gd,&gm, "C:\\TURBOC3\\BCI");

printf("Enter the value of x1 and yl: "); scanf("%f%f",&x1,y1);

printf("Enter the value of x2 and y2: "); scanf("%f%f",&x2,&y2);
dx=als(x2-xl) ;

dy=a~los(y2-y~l)

if(dx>=dy)

step-dx;

else

step=dy;

dx=dx/step;

dy=dy/step;

x=x~1

y=y1

i=1

while(i<=step)

{
```

```
putpixel(x,y,5);

x=x+dx;

y=y+dy

i=i+1

delay(100);

}

closegraph();

}
```

[08:19, 26/03/2024] Sejal Pisal: AIM: Develop the program for Bresenham's Line drawing algorithm.

ST PROGRAM

```
#include<math.h>

#include<graphics.h>

#include<conio.h>

void drawline(int x, int y0, int x1, int y1)

{

int dx, dy, p, x, y;
```

```
dx=x1-x0;

dy=y1-y0;

x=x0;
Animation

y=y0;

n=2^{*}dy-dx;

while(x<x1)

if(p>=0)

{

{

putpixel(x,y,7);

y=y+1;

p=p+2^{}dy-2^{}dx;

#

#

}
else

{

A

putpixel(x.y.7);

p=p+2^{*}dy;

}

x=x+1

}

}

int main()

{

int gdriver=DETECT, gmode, error, x0, y0, x1, y1;

clrscr();

initgraph(&gdriver, &gmode, "c:\\TURBOC3\\bgi");

printf("Enter coordinates of first point: ");

scanf("%f%f",&x 8\times0):
```

```
printf("Enter coordinates of second point: ");

scanf("%f%f",&x1,&y1);

drawline(x0, y0, x1, y1);

closegraph();

}

return 0;
```

[08:20, 26/03/2024] Sejal Pisal: AIM: Develop the program for the mid-point circle drawing algorithm.

AT PROGRAM

```
#include<iostream.h>

#include<graphics.h>

#include<conio.h>

void drawcircle(int x0,int y0,int radius)

{

int x=radius;

int y=0;
int err=0;

while(x>=y)

{ putpixel(x0+x,y0+y,7); putpixel(x0+y,y0+x,7); putpixel(x0-y.y0+x,7); putpixel(x0-x,y0+y,7);
putpixel(x0-x,y0-y.7); putpixel(x0-y,y0-x.7); putpixel(x0+y,y0-x,7); putpixel(x0+x,y0-y,7);

if(err<=0)

{

y+=1;

err+=2*y+1;

}

if(err>=0)

{x-=1;

err-=2*x+1;

}

}

}

int main()
```

```
{

int gddriver DETECT.gmode, error,x,y,r;

initgraph(&gddriver,&gmode, "C:\\TURBOC3\\BGI");

cout<<"Enter radius of circle:":

cin>>r;

cout<<"Enter co-ordinates of center(x&y)";

cin>>x>>y;

drawcircle(x,y,r);

getch();

return 0;

}
```

[08:24, 26/03/2024] Sejal Pisal: AIM: Develop the program for the mid-point ellipse drawing algorithm.

PROGRAM

```
#include<graphics.h>

#include<stdlib.h>

#include<iostream.h>
```

```
#include<conio.h>

void main()

{

clrscr();

int gd DETECT, gm;

int xe,ye,x,y; float p;

long rx,ry;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI"); cout<<"Enter coordinates of centre: ";

cin>>xc>>yc;

cout<<"Enter x,y radius of ellipse: ";

cin>>rx>>ry;

//Region 1

p=ry*ry-rx*rx*ry+rx*rx/4;

x=0;y=ry;

while(2.0*ry*ry*x <= 2.0*rx*rx*y)
```

```
{

if(p < 0)

{

x++;

p = p+2*ry*ry*x+ry*ry;

}

else

{

x++;y--;

p = p+2*ry*ry*x-2*rx*rx*y-ry*ry;

}

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

putpixel(xe-x,ye+y,RED);

putpixel(xe-x,ye-y, RED);

}
```

```
//Region 2

p=ry*ry*(x+0.5)(x+0.5)+rx*rx(y-1)*(y-1)-rx*x*ry*ry; while(y > 0)

{

if(p <= 0)

{

x++;y--;

p=p+2*ry*ry*x-2*rx*rx*y+rx*rx;

}

else

{

y--;

p = p-2*rx*rx*y+rx*rx;

}

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

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

getch();
```

```
closegraph();

}
```

[08:25, 26/03/2024] Sejal Pisal: PRACTICAL NO: 6A

AIM: Write a program to implement 2D scaling.

PROGRAM

```
#include<graphics.h>

#include<stdlib.h>

#include<stdio.h>

#include<math.h>

void main()

{

int graphdriver=DETECT,graphmode,errorcode;

int i;

int x2,y2,x1,y1,x,y;

printf("Enter the 2 line end points:");

printf("x1,y1,x2,y2");
scanf("%d%d%d%d",&x1,y1,&x,&y2);

(&graphdriver,&graphmode, "C:\\TURBOC3\\BGI");

line(x1,y1,x2,y2);

printf("Enter scaling co-ordinates ");

printf("x,y");

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

x1=(x1*x);

y1=(y1*y);

x2=(x2*x);

y2=(y2*y);

printf("Line after scaling");

line(x1,y1,x2,y2);

getch();

closegraph();

}
```
[08:26, 26/03/2024] Sejal Pisal: PRACTICAL NO: 6B

AIM: Write a program to perform 2D translation.

PROGRAM

```
#include<graphics.h>

#include<stdlib.h>

#include<stdio.h>

#include<math.h>

void main()

{

int graphdriver=DETECT,graphmode,errorcode;

int i;

int x2,y2,x1,y1,x,y;

printf("Enter the 2 line end points:");

printf("x1,y1,x2,y2");

scanf("%d%d%d%d",&x1,y1,&x2,y2);

initgraph(&graphdriver,&graphmode, "C:\\TURBOC3\\BGI");
line(x1,y1,x2,y2);

printf("Enter translation co-ordinates ");

printf("x,y");

\\BGI");

0

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

x1=x1+x;

yl=y1+y;

x2=x2+x:

y2=y2+y;

printf("Line after translation");

line(x1,y1,x2,y2);

getch();

closegraph();

}
```

[08:27, 26/03/2024] Sejal Pisal: AIM: Perform 2D Rotation on a given object.

PROGRAM

```
#include<graphics.h>

#include<stdlib.h>

#include<stdio.h>

#include<math.h>

#include<conio.h>

void main()

{

int graphdriver=DETECT,graphmode, errorcode; int i;

int x2,y2,x1,y1,x,y,xn,yn;

double r11,r12,th;

float r21,r22;

clrscr();

printf("Enter the 2 line end points:");

printf("x1,y1,x2,y2 ");

scanf("%d%d%d%d",&x1,y1,&x2,&y2);
initgraph(&graphdriver, &graphmode, "C:\\TURBOC3\\BGI"); line(x1,y1,x2,y2);

printf("\n\n\n Enter the angle ");

scanf("%lf",&th);

r11=cos((th*3.1428)/180);

r12=sin((th*3.1428)/180);

r21=(-sin((th*3.1428)/180));

r22=cos((th*3.1428)/180);

//printf("%lf %lf %lf %lf", r11,r12,r21, r22);

xn=((x2*r11)-(y2*r12));

yn=((x2*r12)+(y2*r11));

line(x1,y1,xn,yn);

getch();

closegraph();

}
```

[08:29, 26/03/2024] Sejal Pisal: PRACTICAL NO: 9A

AIM: Write a program to fill a circle using Flood Fill Algorithm.

PROGRAM

```
#include<stdio.h>

#include<graphics.h>

#include<dos.h>

void floodFill(int x,int y,int oldcolor, int newcolor)

{

if(getpixel(x,y) == oldcolor)

{

putpixel(x,y,newcolor); floodFill(x+1,y,oldcolor,newcolor); flood Fill(x,y+1,oldcolor,newcolor);
floodFill(x-1,y,oldcolor,newcolor); floodFill(x,y-1,oldcolor,newcolor);

}

}

//getpixel(x,y) gives the color of specified pixel int main()

{

int gm,gd=DETECT, radius;

int x,y;
printf("Enter x and y positions for circle\n");

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

printf("Enter radius of circle\n");
scanf("%d",&radius); inityraph(&gl&gm." surtex); circle(z,y radius); flood Fill(x,y 0.15); delay(5000):
closegraph(); return 0; }
```

[08:30, 26/03/2024] Sejal Pisal: PRACTICAL NOOD

AIM: Write a program to fill a circle using Boundary Fill Algorithm.

PROGRAM

```
#include<iostream.h>

#include<graphics.h>

#include<dos.h>

void boundaryfill(int x,int y,int f_color.int b_color)

{

if(getpixel(x,y)!=b_color && getpixel(x,y)!=f_color)

{

putpixel(x,y.f_color);

boundaryfill(x+1,y,f_color.b_color);
boundaryfill(x,y +1,f_color,b_color);

boundaryfill(x-1,y,f_color.b_color);

boundaryfill(x,y-1,f_color,b_color);

}

}

int main()

{
int gm,gd=DETECT, radius;

int x,y;

cout<<"Enter x & y positions for circle \n";

cin>>x>>y;

cout<<"Enter radius of circle \n";

cin>>radius;

initgraph(&gd,&gm, "C:\\TURBOC3\\BGI");

circle(x,y,radius);

boundaryfill(x,y,4,15);
```

```
delay(5000);

closegraph();

return 0;

}
```

[08:32, 26/03/2024] Sejal Pisal: PRACTICAL NO: 10A

AIM: Develop a simple text screen saver using graphics functions.

PROGRAM

```
#include<conio.h>

#include<iostream.h>

#include<graphics.h>

void main()

{

int gd=DETECT,gm,maxx,maxy;

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

maxx=getmaxx()/2;

maxy=getmaxy()/2;
while(!kbhit())
{ for(int i=0;i<maxy;i++)

{

cleardevice();

settextstyle(3,0,5);

outtextxy (maxx/2,i, "Graphics c");

}}

getch();

}
```

[08:33, 26/03/2024] Sejal Pisal: AIM: Perform smiling face animation using graphic functions.

PROGRAM

```
#include <graphics.h>

#include<conio.h>

#include<stdlib.h>

main()

{
```

```
int gd DETECT, gm, area, temp1, temp2, left = 25,

75:

тор



void *p;

initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");

setcolor(YELLOW);

circle(50, 100, 25);

setfillstyle(SOLID_FILL, YELLOW);

floodfill (50, 100, YELLOW);

setcolor(BLACK);

setfillstyle(SOLID_FILL, BLACK);

fillellipse(44, 85, 2, 6);

fillellipse (56, 85, 2, 6);

ellipse(50, 100, 205, 335, 20, 9);

ellipse(50, 100, 205, 335, 20, 10);
ellipse(50, 100, 205, 335, 20, 11);

area imagesize(left, top, left + 50, top + 50); pmalloc(area); setcolor(WHITE);
settextstyle(SANS_SERIF_FONT, HORIZ_DIR, 2); outtextxy (155, 451, "Smiling Face Animation");
setcolor(BLUE); rectangle(0, 0, 639, 449); while(!kbhit()) {

temp11+ random (588); temp21+ random (380); getimage(left, top, left + 50, top + 50, p):
putimage(left, top. p. XOR_PUT); putimage(templ, temp2. p, XOR_PUT); delay(100); left = templ; top
= temp2; }

getch();

closegraph();

return 0;
```

0

Edit

All

Annotate

X

Fill & Sign

}