DATA STRUCTURES AND ALGORITHMS COURSE PROJECT

SUBMITTED BY:

**SHRIYA** 21071A3226

**SIRI** 21071A3227

***JASHWANTH*** 21071A3228

**SUMITHA** 21071A3229

***J.VISHWAJITHA*** 21071A3230

UNDER THE GUIDANCE OF **DR.SABBINENI NAGINI**

BRANCH : COMPUTER SCIENCE AND BUSINESS SYSTEMS DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

# VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with A++ Grade NBA Accredited for CE, EEE, ME, ECE,

CSE, EIE, IT B. Tech Courses Approved by AICTE, New Delhi, Affiliated to JNTUH Recognized as

"College with Potential for Excellence" by UCC EAMCET CODE VJEC, PGECET CODE: VJEC 1

**Course project 1: Program to print the names and roll numbers of our team members in a different shape with different colors.**

CODE:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main ()

{

int gdriver=DETECT,gmode;

initgraph(&gdriver,&gmode,"c:\\turboc3\\bgi");

setcolor(4);

circle(60,100,60);

setfillstyle(1,1);

outtextxy(40,100,"shriya");

outtextxy(40,170,"26");

floodfill(61,101,4);

rectangle(120,40,200,150);

setfillstyle(1,10);

outtextxy(140,100,"siri");

outtextxy(145,170,"27");

floodfill(121,41,4);

circle(260,100,60);

setfillstyle(1,11);

outtextxy(220,100,"jaswanth");

outtextxy(240,170,"28");

floodfill(261,101,4);

rectangle(320,40,400,150);

setfillstyle(1,5);

outtextxy(330,100,"sumitha");

outtextxy(350,170,"29");

floodfill(331,101,4);

circle(460,100,60);

setfillstyle(1,14);

outtextxy(410,100,"vishwajitha");

outtextxy(450,170,"30");

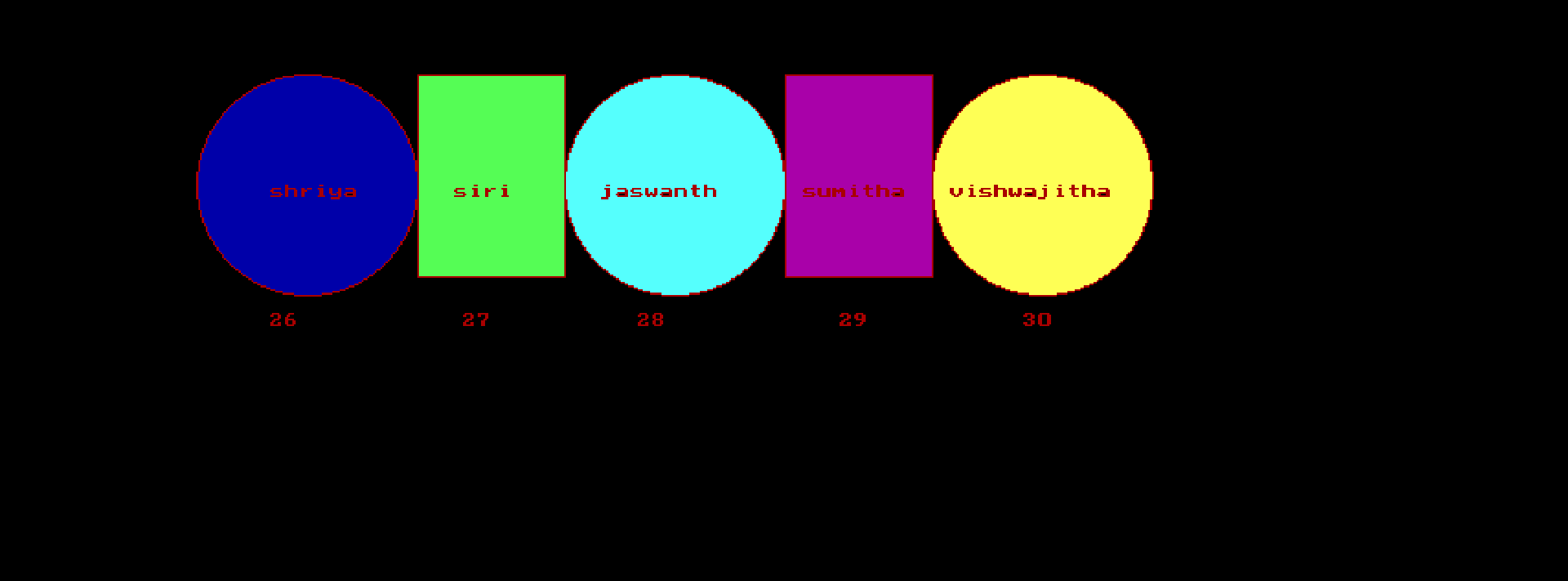
floodfill(411,101,4);

getch();

closegraph();

}

OUTPUT:



**Course project 2:**

**Simulation of dequeue by graphics:**

**Code:**

#include<conio.h>

#include<graphics.h>

#include<ctype.h>

#include<dos.h>

#include<string.h>

#include<stdio.h>

//functions

void graphicspointer(int);

void showindex(void);

void graphicsinsertrear(int);

void graphicsinsertfront(int);

void graphicsdeletrear(int);

void graphicsdeletfront(int);

void graphicsboxmax(int);

void graphicsdefault(void);

char title[50];

int f=0,r=-1,max,ele,j=0,d=0,i,p,flag;

int dq[100];

void insertfront(void);

void insertrear(void);

void deletfront(void);

void deletrear(void);

void main()

{

int gd=DETECT,gm,l,p;

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

cleardevice();

settextstyle(0,0,0);

setbkcolor(0);

strcpy(title,"DEQUEUE SIMULATION");

gotoxy(30,2);

printf("%s",title);

printf("\n\n\nEnter the maximum number of elements-\n");

scanf("%d",&max);

cleardevice();

graphicsdefault();

}

void graphicsdefault(void)

{

strcpy(title,"DEQUEUE SIMULATION");

gotoxy(30,2);

printf("%s",title);

printf("\n\nHere's your Deueue\n");

graphicsboxmax(max);

graphicspointer(max);

setcolor(15);

printf("\nOperations : 1.Insertfront 2.insertrear 3.deletfront 4.deletrear 5.Exit\n\n\n\n\n\n\n\n");

do

{

setfillstyle(SOLID\_FILL,0);

bar(0,192,1000,800);

scanf("%d",&p);

switch(p)

{

case 1: if(f!=0)

{

insertfront();

}

else

{

printf("insertfront not feasible\n");

getch();

}

break;

case 2:if(r>=max-1)

{

printf("insertion from rear not possible as queue is full\n");

}

else

{

insertrear();

getch();

}

break;

case 3: if(r!=-1)

deletfront();

else

{

printf("deletion from front not possible\n");

sleep(2);

}

break;

case 4: if(r!=-1)

deletrear();

else

{

printf("deletion from rear not possible\n");

sleep(2);

}

break;

case 5: break;

}

}while(p!=5);

}

void graphicsboxmax(int n)

{

int i,x1;

x1=0;

for(i=1;i<=n;i++)

{

rectangle(x1,150,x1+60,180);

x1=x1+60;

}

}

void graphicspointer(int n)

{

int i,x1;

x1=0;

for(i=1;i<=n;i++)

{

setcolor(0);

rectangle(x1,130,x1+60,140);

x1=x1+60;

}

x1=0;

for(i=1;i<=n;i++)

{

setcolor(0);

rectangle(x1,120,x1+60,130);

x1=x1+60;

}

}

void insertfront(void)

{

printf("Enter the element to insert: ");

scanf("%d",&ele);

f=f-1;

dq[f]=ele;

graphicsinsertfront(ele);

sleep(3);

graphicsdefault();

}

void graphicsinsertfront(int a)

{

sprintf(title,"%d",a);

setfillstyle(SOLID\_FILL,d);

bar(60\*(d-1),151,60\*(d)-1,179);

outtextxy(30+60\*(d-1),160,title);

outtextxy(30+60\*(d-1),124,"f");

setfillstyle(SOLID\_FILL,0);

bar(60\*d,121,60\*(d+1),129);

d=d-1;

}

void insertrear(void)

{

printf("Enter the element to insert: ");

scanf("%d",&ele);

r=r+1;

dq[r]=ele;

graphicsinsertrear(ele);

sleep(3);

graphicsdefault();

}

void graphicsinsertrear(int a)

{

sprintf(title,"%d",a);

setfillstyle(SOLID\_FILL,j+1);

bar(60\*j+1,151,60\*(j+1)-1,179);

outtextxy(30+60\*j,160,title);

showindex();

if(f==0&&r!=max)

{

outtextxy(30+60\*d,124,"f");

setfillstyle(SOLID\_FILL,0);

bar(60\*(d-1)+1,121,60\*d,129);

setfillstyle(SOLID\_FILL,0);

bar(60\*(j-1)+1,131,60\*j,139);

//rear pointer

outtextxy(30+60\*j,134,"r");

}

j=j+1;

}

void showindex(void)

{

sprintf(title,"%d",j);

outtextxy(30+60\*j,185,title);

}

void deletfront(void)

{

int x;

flag=r+f;

if(f==r)

{

x=dq[f];

f=0;

r=-1;

}

else

{

x=dq[f];

f=(f+1);

}

graphicsdeletfront(x);

graphicsdefault();

}

void graphicsdeletfront(int x)

{

setfillstyle(SOLID\_FILL,0);

bar(60\*d+1,151,60\*(d+1)-1,179);

outtextxy(30+60\*(d+1),124,"f");

setfillstyle(SOLID\_FILL,0);

bar(60\*d,121,60\*(d+1),129);

d=d+1;

printf("%d is deleted\n(Enter any key to continue)\n",x);

getch();

}

void deletrear(void)

{

int x;

flag=r+f;

if(f==r)

{

x=dq[f];

f=0;

r=-1;

}

else

{

x=dq[r];

r=(r-1);

}

graphicsdeletrear(x);

graphicsdefault();

}

void graphicsdeletrear(int x)

{

setfillstyle(SOLID\_FILL,0);

bar(60\*(j-1),151,60\*j,179);

outtextxy(30+60\*(j-2),134,"r");

j=j-1;

setfillstyle(SOLID\_FILL,0);

bar(60\*(j),131,60\*(j+1),139);

printf("%d is deleted\n(Enter any key to continue)\n",x);

getch();

}

**Output:**





