***SOURCE CODE***

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<string.h>

#include<graphics.h>

#include<dos.h>

int front=-1,rear=-1,q[3][10];

int full,total;

void enqueue(char \*);

struct nod1

{ char name1[50];

char number[10];

struct nod1 \*next;

};

typedef struct nod1 node;

struct nod

{ char alphabet;

struct nod \*down;

struct nod1 \*right;

};

typedef struct nod headnode;

typedef struct name

{

char n[10];

struct name \*d;

struct nod \*r;

}name;

node \* frenz();

headnode \* alpha();

void main()

{

name \*s1=NULL,\*ptr,\*temp;

void search(headnode \*);

void viewfrenz(name \*);

void view(name \*);

void maximum(name \*);

name \* insertname(name \*);

name \* insertfriend(name \*);

void qdisplay();

int choice=1,a,choice2=1;

int x,y,i;

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\Turboc3\\BGI");

setcolor(GREEN);

settextstyle(4,0,8);

outtextxy(200,50,"PHONE");

setcolor(YELLOW);

settextstyle(4,0,8);

outtextxy(100,190,"DIRECTORY");

settextstyle(1,0,1);

setcolor(WHITE);

settextstyle(50,0,1);

outtextxy(380,400,"Shubham,Vidit,Ankur,Lakshay");

outtextxy(410,420,"B.tech I Year");

getch();

while(!kbhit())

{

x=random(640);

y=random(480);

setcolor(15);

for(i=1;i<10;i++)

{

circle(x,y,i);

delay(100);

}

setfillstyle(1,15);

line(x+8,y-2,x+40,y);

line(x+8,y+2,x+40,y);

floodfill(x+11,y,15);

line(x-8,y-2,x-40,y);

line(x-8,y+2,x-40,y);

floodfill(x-11,y,15);

line(x-2,y+8,x,y+40);

line(x+2,y+8,x,y+40);

floodfill(x,y+11,15);

line(x-2,y-8,x,y-40);

line(x+2,y-8,x,y-40);

floodfill(x,y-11,15);

line(x+8,y-2,x+20,y-20);

line(x+2,y-8,x+20,y-20);

floodfill(x+15,y-15,15);

line(x+8,y+2,x+20,y+20);

line(x+2,y+8,x+20,y+20);

floodfill(x+15,y+15,15);

line(x-8,y+2,x-20,y+20);

line(x-2,y+8,x-20,y+20);

floodfill(x-15,y+15,15);

line(x-8,y-2,x-20,y-20);

line(x-2,y-8,x-20,y-20);

floodfill(x-15,y-15,15);

sound(4000);

setcolor(0);

for(i=40;i>=10;i--)

{

line(x+8,y-2,x+i,y);

line(x+8,y+2,x+i,y);

}

for(i=40;i>=10;i--)

{

line(x-8,y-2,x-i,y);

line(x-8,y+2,x-i,y);

}

for(i=40;i>=10;i--)

{

line(x-2,y+8,x,y+i);

line(x+2,y+8,x,y+i);

}

for(i=40;i>=10;i--)

{

line(x-2,y-8,x,y-i);

line(x+2,y-8,x,y-i);

}

for(i=20;i>=7;i--)

{

line(x+8,y-2,x+i,y-i);

line(x+2,y-8,x+i,y-i);

}

for(i=20;i>=7;i--)

{

line(x+8,y+2,x+i,y+i);

line(x+2,y+8,x+i,y+i);

}

for(i=20;i>=7;i--)

{

line(x-8,y+2,x-i,y+i);

line(x-2,y+8,x-i,y+i);

}

for(i=20;i>=7;i--)

{

line(x-8,y-2,x-i,y-i);

line(x-2,y-8,x-i,y-i);

}

for(i=9;i>0;i--)

{

circle(x,y,i);

delay(10);

}

nosound();

}

cleardevice();

getch();

closegraph();

while(choice==1)

{

clrscr();

printf("\n\t\t\t\tMENU\n");

printf("\t\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

printf("\t\t1. create phone directory\n");

printf("\t\t2. view directory\n");

printf("\t\t3. to show friends with initial alphabet\n");

printf("\t\t4. to insert a new name\n");

printf("\t\t5. to insert a friend at a given alphabet's list\n");

printf("\t\t6. to see who has maximum friends\n");

printf("\t\t7. display\n");

printf("\t\t8. Exit\n\n");

printf("\t\tEnter the choice: ");

scanf("%d", &a);

switch(a)

{

case 1:

choice2=1;

while(choice2==1)

{

clrscr();

ptr=(name \*)malloc(sizeof(name));

ptr->d=NULL;

printf("enter the name: ");

scanf("%s",ptr->n);

enqueue(ptr->n);

getch();

ptr->r=alpha();

if(s1==NULL)

{

s1=ptr;

}

else

{

temp=s1;

while(temp->d!=NULL)

{

temp=temp->d;

}

temp->d=ptr;

ptr->d=NULL;

}

clrscr();

printf("\ndo you want more name?? 1 yes/2 no\n");

scanf("%d", &choice2);

}

break;

case 2: view(s1); break;

case 3: viewfrenz(s1); break;

case 4: s1=insertname(s1); break;

case 5: s1=insertfriend(s1); break;

case 6: maximum(s1); break;

case 7: qdisplay(); break;

case 8: exit(0); break;

default: printf("Wrong choice");

}

clrscr();

printf("\ndo you want to continue?? 1 yes/ 0 no\n");

scanf("%d", &choice);

}

getch();

}

headnode \*alpha()

{

headnode \*start1=NULL,\*temp1,\*ptr1;

int choice2=1,z;

while(choice2==1)

{

clrscr();

ptr1=(headnode \*)malloc(sizeof(headnode));

ptr1->down=NULL;

printf("\nenter the alphabet: ");

ptr1->alphabet=getche();

getch();

ptr1->right=frenz();

if(start1==NULL)

{

start1=ptr1;}

else

{

temp1=start1;

while(temp1->down!=NULL)

{

temp1=temp1->down;

}

temp1->down=ptr1;

ptr1->down=NULL;

}

clrscr();

printf("\ndo you want to add more alphabets?? 1 yes/ 0 no\n");

scanf("%d", &choice2);

}

return(start1);

}

node \*frenz()

{

node \*start2=NULL,\*temp2,\*ptr2;

int choice2=1,z;

while(choice2==1)

{

clrscr();

ptr2=(node \*)malloc(sizeof(node));

ptr2->next=NULL;

printf("\nenter the name of friend\n");

scanf("%s", &ptr2->name1);

printf("enter the mobile number\n");

scanf("%s", &ptr2->number);

if(start2==NULL)

{

start2=ptr2;}

else

{

temp2=start2;

while(temp2->next!=NULL)

{

temp2=temp2->next;

}

temp2->next=ptr2;

}

clrscr();

printf("\ndo you want to add more friends?? 1 yes/ 0 no\n");

scanf("%d", &choice2);

}

return(start2);

}

void view(name \*start)

{

name \*temp2;

headnode \*temp1;

node \*temp;

temp2=start;

clrscr();

while(temp2!=NULL)

{

printf("\nname: %s\n",temp2->n);

temp1=temp2->r;

while(temp1!=NULL)

{

printf("\nalphabet: %c\n", temp1->alphabet);

temp=temp1->right;

while(temp!=NULL)

{

printf("friends: %s\n",temp->name1);

printf("mobile number: %s\n",temp->number);

temp=temp->next;

}

printf("------------------------------------------------------------");

temp1=temp1->down;

}

temp2=temp2->d;

}

getch();

}

void viewfrenz(name \*start)

{

char alp;

char na[10];

name \*temp2;

headnode \*temp;

node \*temp1;

clrscr();

printf("\nenter the name: ");

scanf("%s",&na);

temp2=start;

while(strcmp(na,temp2->n)!=0)

{

temp2=temp2->d;

}

temp=temp2->r;

printf("\nenter the alphabet: ");

alp=getche();

while(alp!=temp->alphabet)

{

temp=temp->down;

}

temp1=temp->right;

printf("\nalphabet: %c",temp->alphabet);

while(temp1!=NULL)

{

printf("\nFriend: %s", &temp1->name1);

printf("\nMobile number: %s", &temp1->number);

temp1=temp1->next;

}

getch();

}

name \* insertname(name \* start)

{

name \*temp, \*ptr;

clrscr();

ptr=(name \*)malloc(sizeof(name));

printf("\nenter the name: ");

scanf("%s",ptr->n);

ptr->r=alpha();

ptr->d=NULL;

temp=start;

while(temp->d!=NULL)

{ temp=temp->d;

}

temp->d=ptr;

clrscr();

printf("your work has been done\n");

getch();

return(start);

}

name \* insertfriend(name \*start1)

{

name \*temp1;

headnode \*temp2;

node \*temp, \*ptr, \*start;

char s;

char na[10];

clrscr();

printf("\nenter the name in which friend is to be added: ");

scanf("%s",&na);

temp1=start1;

while(strcmp(na,temp1->n)!=0)

{

temp1=temp1->d;

}

temp2=temp1->r;

printf("\nenter the alphabet in which friend is to be added: ");

s=getche();

while(s!=temp2->alphabet)

{

temp2=temp2->down;

}

start=temp2->right;

temp=start;

while(temp->next!=NULL)

{

temp=temp->next;

}

ptr=(node \*)malloc(sizeof(node));

printf("\nenter the name of friend\n");

scanf("%s", &ptr->name1);

printf("\nenter the mobile number\n");

scanf("%s", &ptr->number);

ptr->next=NULL;

temp->next=ptr;

printf("\nyour work has been done");

return(start1);

}

void enqueue(char n[])

{

int x;

int max=3;

if(total==3)

{

total=0;

full=1;

}

else

full=1;

printf("Added : %s",n);

if(front==0&&rear==max-1)||(front==rear+1)) //condition for full Queue

{

rear=0;

front++; printf("Queue is overflow\n");return;

}

else if(front==rear+1)

{

rear++;

front++;

}

else if(front==-1)

{

front=rear=0;

}

else

{

rear++;

}

strcpy(q[total],n);

total++;

printf("asdasa : %s\n",q[0]);

getch();

printf("enter the no:\n");

scanf("%d",&x);

q[rear]=x;

printf("%d succ. inserted\n",x);

return;

}

void maximum(name \*s1)

{

char arr[10];

int max,count;

char s;

name \*temp2;

headnode \*temp,\*start;

node \*temp1,\*start1;

temp2=s1;

clrscr();

max=0;

count=0;

while(temp2!=NULL)

{

start=temp2->r;

temp=start;

while(temp!=NULL)

{

start1=temp->right;

temp1=start1;

count=0;

while(temp1!=NULL)

{

count++;

temp1=temp1->next;

}

if(max<=count)

{

max=count;

strcpy(arr,temp2->n);

s=temp->alphabet;

}

temp=temp->down;

}

temp2=temp2->d;

}

printf("name: %s",arr);

printf("\n%c: friends:%d", s,max);

getch();

}

void qdisplay(void)

{

int i,j;

int max=3;

clrscr();

if(front==rear==-1)

{

printf("q is empty\n");return;

}

printf("Names are :\n");

for(i=front;i!=rear;i=(i+1)%max)

{

printf("%s\n",q[i][10]);

}

printf("%s\n",q[rear][10]);

if(full==1)

total=3;

j=0;

printf("Names are :\n");

for(i=0;i<3;i++)

{

printf("%s\n",q[i]);

}

getch();

return;

}