#include<stdio.h>

#include<conio.h>

#include<string.h>

#include<stdlib.h>

#define max 4

struct data

{

char job[max];

int prno;

int ord;

};

struct queue

{

struct data que[max];

int front,rear;

};

typedef struct queue queue;

void initQueue(queue \*q) //initialise front and rear

{

q->front=q->rear=-1;

}

void insertion(queue \*q,struct data item)

{

int i,j;

struct data temp;

if(q->rear==max-1) //check for queue full condition

{

printf("\nOVERFLOW");

return;

}

q->rear++;

if(q->front==-1) //first element is being inserted

q->front=0;

q->que[q->rear]=item;

for(i=q->front;i<q->rear;i++)

{

for(j=q->front;j<q->rear-i;j++)

{

if(q->que[j].prno>q->que[j+1].prno)

{

temp=q->que[j];

q->que[j]=q->que[j+1];

q->que[j+1]=temp;

}

{

if(q->que[j].prno==q->que[j+1].prno)

{

if(q->que[j].ord>q->que[j+1].ord)

{

temp=q->que[j];

q->que[j]=q->que[j+1];

q->que[j+1]=temp;

}

}

}

}

}

}

struct data deletion(queue \*q)

{

struct data item;

strcpy(item.job," ");

item.prno=-32768;

item.ord=-32768;

if(q->front==-1) //queue is empty

{

printf("\nUnderflow");

return(item);

}

item=q->que[q->front];

if(q->front==q->rear) //only one element present

q->front=q->rear=-1;

else

q->front++;

return(item);

}

void display(queue \*q)

{

int i;

for(i=q->front;i<=q->rear;i++)

{

printf("\n%s %d %d ",q->que[i].job,q->que[i].prno,q->que[i].ord);

}

fflush(stdin);

getchar();

}

int search(queue \*q,char item[])

{

int i;

for(i=q->front;i<=q->rear;i++)

{

if(strcmp(q->que[i].job,item)==0)

return(i);

}

return(max);

}

void main()

{

int ch,pos,i=1;

struct data item;

char s[10];

queue q;

initQueue(&q);

while(1)

{

system("cls");

printf("\n1.Insert 2.Delete 3.Search 4.Display 5.Exit\n");

scanf("%d",&ch);

if(ch==5)

break;

switch(ch)

{

case 1:

printf("\nEnter element and its priority you want to insert\n");

scanf("%s %d",item.job,&item.prno);

item.ord=i++;

insertion(&q,item);

break;

case 2:

item=deletion(&q);

printf("\nDeleted item is %s with priority %d",item.job,item.prno);

break;

case 3:

printf("\nEnter element you want to search\n");

scanf("%s",s);

pos=search(&q,s);

if(pos==max)

printf("\nSorry Element not found");

else

printf("\nElement %s is present at %d position",item.job,pos+1);

break;

case 4:

printf("\nElements in queue are -> ");

display(&q);

break;

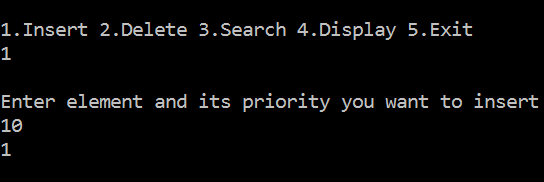
default:

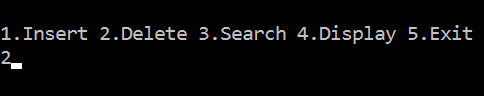
printf("\nInvalid choice");

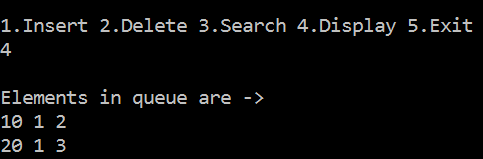
}

}

}







#include<stdio.h>

#include<stdlib.h>

struct node

{

int priority;

int info;

struct node \*link;

}\*front=NULL;

void insert(int item, int item\_priority);

int del();

void display();

int isEmpty();

void main()

{

int choice,item,item\_priority;

while(1)

{

system("cls");

display();

printf("1.Insert\n");

printf("2.Delete\n");

printf("3.Display\n");

printf("4.Quit\n");

printf("Enter your choice : ");

scanf("%d", &choice);

switch(choice)

{

case 1:

printf("Input the item to be added in the queue : ");

scanf("%d",&item);

printf("Enter its priority : ");

scanf("%d",&item\_priority);

insert(item, item\_priority);

break;

case 2:

printf("Deleted item is %d\n",del());

break;

case 3:

display();

break;

case 4:

exit(1);

default :

printf("Wrong choice\n");

}

}

}

void insert(int item,int item\_priority)

{

struct node \*tmp,\*p;

tmp=(struct node \*)malloc(sizeof(struct node));

if(tmp==NULL)

{

printf("Memory not available\n");

return;

}

tmp->info=item;

tmp->priority=item\_priority;

if( isEmpty() || item\_priority < front->priority )

{

tmp->link=front;

front=tmp;

}

else

{

p = front;

while( p->link!=NULL && p->link->priority<=item\_priority )

p=p->link;

tmp->link=p->link;

p->link=tmp;

}

}

int del()

{

struct node \*tmp;

int item;

if( isEmpty() )

{

printf("Queue Underflow\n");

exit(1);

}

else

{

tmp=front;

item=tmp->info;

front=front->link;

free(tmp);

}

return item;

}

int isEmpty()

{

if( front == NULL )

return 1;

else

return 0;

}

void display()

{

struct node \*ptr;

ptr=front;

if( isEmpty() )

printf("Queue is empty\n");

else

{

printf("Queue is :\n");

printf("Priority Item\n");

while(ptr!=NULL)

{

printf("%5d %5d\n",ptr->priority,ptr->info);

ptr=ptr->link;

}

}

}

