/\*quque using linked list\*/

#include<stdio.h>

#include<stdlib.h>

struct queue

{

int data;

struct queue\*link;

};

struct queue\*front;

struct queue\*rear;

void insert(int item);

void delete\_();

void display();

int main()

{

int ch,item;

while(1)

{

printf("\*\*main menu\*\*\n");

printf("1 - insert\n");

printf("2 -delete\n");

printf("3 -display\n");

printf("4 -exit\n");

printf("enter your choice: \n");

scanf("%d",&ch);

switch(ch)

{

case 1:

printf("enter the no to be inserted in queue: \n");

scanf("%d",&item);

insert(item);

break;

case 2:

delete\_();

break;

case 3:

display();

break;

case 4:

exit(0);

default:

printf("invalid choice\n");

}

}

}

void insert(int item)

{

struct queue\*new\_node;

if(new\_node==NULL)

{

printf("\nOVERFLOW\n");

return;

}

else

{

new\_node=(struct queue\*) malloc(sizeof(struct queue\*));

new\_node->data=item;

if(front==NULL)

{

front=new\_node;

rear=new\_node;

front->link=NULL;

rear->link=NULL;

}

else

{

rear->link=new\_node;

rear=new\_node;

rear->link=NULL;

}

printf("item inserted\n");

}

}

void delete\_()

{

struct queue\*ptr;

if(front==NULL)

{

printf("queue is empty\n");

return;

}

else

{

ptr=front;

front= front->link;

printf("deleted value: %d\n",ptr->data);

free(ptr);

}

/\*

other code

{

else

{

int x;

x=front->data;

front=front->link;

printf("the deleted value is %d",x);

}

}

\*/

}

void display()

{

struct queue\*ptr;

ptr=front;

if(front==NULL)

{

printf("quque is empty\n");

}

else

{ printf("queue is below\n");

while(ptr!= NULL)

{

printf("%d\n",ptr->data);

ptr=ptr->link;

}

}

}

