/\*quque using linked list\*/

#include<stdio.h>

#include<stdlib.h>

struct queue

{

int data;

struct queue\*link;

};

struct queue\*front; //here,header=front

struct queue\*insert\_queue(struct queue\*,int);

struct queue\*delete\_queue(struct queue\*);

struct queue\*display\_queue(struct queue\*);

int main()

{

int item,ch;

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);

front=insert\_queue(front,item);

break;

case 2:

front=delete\_queue(front); //return the rest of list after deleting a data

break;

case 3:

front=display\_queue(front);

break;

case 4:exit(0);

default:

printf("invalid choice\n");

}

}

}

struct queue\*insert\_queue(struct queue\*front,int item)

{

struct queue\*new\_node,\*rear;

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");

return front;

}

struct queue\*delete\_queue(struct queue\*front)

{

struct queue\*rear,\*ptr;

if(front==NULL)

{

printf("queue is empty\n");

}

else

{

ptr=front;

front=front->link;

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

free(ptr);

}

return front;

/\*

other code

{

else

{

int x;

x=front->data;

front=front->link;

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

}

}

\*/

}

struct queue\*display\_queue(struct queue\*front)

{

struct queue\*rear,\*ptr;

if(front==NULL)

{

printf("queue is empty\n");

}

else

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

ptr=front;

while(ptr!=NULL)

{

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

ptr=ptr->link;

}

}

return front;

}