//circular queue implementation using linked list

#include<stdio.h>

#include<malloc.h>

struct node

{

int data;

struct node \*next;

};

typedef struct node node;

//global declaration of front and rear pointer

node \*front=NULL;

node \*rear=NULL;

//function declaration

void insert(int);

void del();

void display();

int main()

{

int c=1;

while(c!=0) //menu driven program

{

printf("\n1.Insert item in the queue\t2.Delete item in queue \n3.Display\t4.Peek element\n0.End\n");

scanf("%d",&c);

switch(c)

{

case 1:

{

int val;

printf("Enter the value to be inserted\n");

scanf("%d",&val);

insert(val);

}

break;

case 2:

{

del();

}

break;

case 3:

{

display();

}

break;

case 4:

{

peek();

}

break;

case 0:printf("Exiting");

}

}

return 0;

}

//inserts value at the rear end

void insert(int val)

{

node \*tmp;

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

tmp->next=NULL;

tmp->data=val;

if(front==NULL&&rear==NULL)//when queue is empty

{

front=rear=tmp;

rear->next=tmp;

printf("%d has been inserted in the queue\n",val);

return;

}

else

{

if(front->next==front)//when queue has only one node

{

front->next=tmp;

rear=tmp;

rear->next=front;

printf("%d has been inserted in the queue\n",val);

return;

}

tmp->next=front;

rear->next=tmp;

rear=tmp;

return;

}

}

//deletes from the front end

void del()

{

if(front==NULL&&rear==NULL)//when queue is empty

{

printf("The queue is empty\n");

return;

}

else

{

node \*tmp;

if(front->next==front)//when queue has only one node

{

tmp=front;

printf("%d has been deleted from queue\n",tmp->data);

free(tmp);

front=rear=NULL;

return;

}

tmp=front;

front=front->next;

rear->next=front;

printf("%d has been deleted from the queue\n",tmp->data);

free(tmp);

return;

}

}

void display()

{

if(front==NULL&&rear==NULL)//when queue is empty

{

printf("The queue is empty\n");

return;

}

else

{

node \*ptr;

ptr=front;

printf("front--->");

do

{

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

ptr=ptr->next;

}

while(ptr!=front);

printf("rear\n\n");

return;

}

}

//prints the peek element in the queue

void peek()

{

if(front==NULL&&rear==NULL)//when queue is empty

{

printf("The queue is empty\n");

return;

}

printf("Peek element is %d\n",front->data);

return;

}

OUTPUT:

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

2

The queue is empty

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

3

The queue is empty

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

4

The queue is empty

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

1

Enter the value to be inserted

10

10 has been inserted in the queue

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

3

front--->10--->rear

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

1

Enter the value to be inserted

20

20 has been inserted in the queue

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

3

front--->10--->20--->rear

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

1

Enter the value to be inserted

30

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

3

front--->10--->20--->30--->rear

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

4

Peek element is 10

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

2

10 has been deleted from the queue

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

2

20 has been deleted from the queue

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

2

30 has been deleted from queue

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

2

The queue is empty

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End

3

The queue is empty

1.Insert item in the queue 2.Delete item in queue

3.Display 4.Peek element

0.End