//21co24 ETHAN MENEZES COMP A

#include<stdio.h>

#include<malloc.h>

struct node

{

int info;

struct node\* next;

};

typedef struct node node;

node \*front,\*rear; //global declaration of front and rear pointer

void display(); //function declaration

void insert(int);

void del();

void peek();

int main()

{

front=NULL;

rear=NULL;

int c;

while(c!=0) //menu to handle the various options

{

printf("\n1.Push element in the queue\n2.Pop element from the queue\n3.Display queue\n4.Peek element in the queue\n");

scanf("%d",&c);

switch(c)

{

case 1:

{

int elem;

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

scanf("%d",&elem);

insert(elem);

}

break;

case 2:

{

del();

}

break;

case 3:display();

break;

case 4:peek();

break;

case 0:printf("Exiting");

}

}

}

void insert(int item) //inserts the element at the rear of the queue

{

node \*tmp;

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

tmp->info=item;

tmp->next=NULL;

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

{

front=rear=tmp;

printf("%d has been inserted at the rear\n",item);

return;

}

rear->next=tmp;

rear=tmp;

printf("%d has been inserted at the rear\n",item);

return;

}

void del() //deletes the element at the front of the queue

{

node \*tmp;

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

{

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

return;

}

tmp=front;

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

front=front->next;

if(front==NULL)

{

rear=NULL;

}

free(tmp);

return;

}

void display() //displays the elements in the queue

{

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

{

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

return;

}

node \*ptr=front;

printf("Front--->");

while(ptr!=NULL)

{

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

if(ptr->next!=NULL)

printf("--->");

ptr=ptr->next;

}

printf("<---Rear\n\n");

return;

}

void peek() //printf the peek element of the queue

{

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

{

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

return;

}

printf("%d is the peek element \n",front->info);

return;

}

**OUTPUT:**

/tmp/UXvHK6ICej.o

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

2

The queue is empty

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

3

The queue is empty

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

4

The queue is empty

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

1

Enter the element to be inserted

10

10 has been inserted at the rear

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

1

Enter the element to be inserted

12

12 has been inserted at the rear

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

1

Enter the element to be inserted

14

14 has been inserted at the rear

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

3

Front--->10--->12--->14<---Rear

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

2

10 has been deleted from the queue

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

2

12 has been deleted from the queue

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

4

14 is the peek element

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

2

14 has been deleted from the queue

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue

2

The queue is empty

1.Push element in the queue

2.Pop element from the queue

3.Display queue

4.Peek element in the queue