//Queue using linked list

#include<stdio.h>

#include<conio.h>

#include<malloc.h>

#include<stdlib.h>

#define SIZE 5

struct queue

{

int data;

struct queue \*link;

};

struct queue \*FRONT=NULL, \*REAR=NULL;

void insert()

{

int count = 0;

struct queue \*temp;

//To count the nodes in the queue

temp = FRONT;

while(temp!=NULL)

{

count = count +1;

temp = temp->link;

}

if(count == SIZE) //logical overflow condition

{

printf("\nQueue overflow!can't insert ");

}

else

{

//New node creation

temp=(struct queue \*)malloc(sizeof(struct queue));

if(temp==NULL)

{

printf("\nMemory is full");

getch();

return;

}

//Read value to insert

printf("\nEnter item to insert:");

scanf("%d",&temp->data);

temp->link= NULL;

//updation of FRONT and REAR after insertion

if(REAR == NULL) //Empty queue

{

REAR = temp;

FRONT = temp;

}

else // At least one node is there in the queue

{

REAR->link= temp;

REAR = REAR->link;

}

}

}

void del\_que()

{

struct queue \*temp;

if(FRONT == NULL)

{

printf("\nQueue underflow!no item to delete");

}

else

{

if(FRONT->link==NULL)

{

printf("\nDeleted item is %d ",FRONT->data);

free(FRONT);

FRONT = NULL;

REAR = NULL;

}

else

{

printf("\nDeleted item is %d ",FRONT->data);

temp=FRONT;

FRONT = FRONT->link;

free(temp);

}

}

}

void display()

{

struct queue \*temp;

if(FRONT == NULL)

{

printf("\nQueue underflow!no item to delete");

}

else

{

printf("\nThe items in the queue are\n");

temp=FRONT;

while(temp!=NULL)

{

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

temp=temp->link;

}

}

}

void main()

{

int ch;

clrscr();

while(1)

{

printf("\n1. Insert\n2. Delete\n3. Display\n4. Exit\n");

printf("Enter your choice: ");

scanf("%d",&ch);

switch(ch)

{

case 1: insert();

break;

case 2: del\_que();

break;

case 3: display();

break;

case 4: exit(0);

default: printf("\nWronf choice: ");

}

}

}