Ex 7: Implementation of Queue using Linked List

Name.: Rowin.U

Reg No.:231801141

PROGRAM :

#include<stdio.h >

#include<malloc.h >

struct queue

{

int data;

struct queue \*next;

};

struct queue \*addq(struct queue \*front);

struct queue \*delq(struct queue \*front);

void main()

{

struct queue \*front;

int reply,option,data;

front=NULL;

do

{

printf("\n1.addq");

printf("\n2.delq");

printf("\n3.exit");

printf("\nSelect the option");

scanf("%d",&option);

switch(option)

{

case 1 : //addq

front=addq(front);

printf("\n The element is added into the queue");

break;

case 2 : //delq

front=delq(front);

break;

case 3 : exit(0);

}

}while(1);

}

struct queue \*addq(struct queue \*front)

{

struct queue \*c,\*r;

//create new node

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

if(c==NULL)

{

printf("Insufficient memory");

return(front);

}

//read an insert value from console

printf("\nEnter data");

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

c->next=NULL;

if(front==NULL)

{

front=c;

}

else

{

//insert new node after last node

r=front;

while(r->next!=NULL)

{

r=r->next;

}}

return(front);

}

struct queue \*delq(struct queue \*front)

{

struct queue \*c;

if(front==NULL)

{

printf("Queue is empty");

return(front);

}

//print the content of first node

printf("Deleted data:%d",front->data);

//delete first node

c=front;

front=front->next;

free(c);

return(front);

}

OUTPUT:

