WEEK-06

Imlpementation of queues using linked list

#include<stdio.h>

#include<stdlib.h>

struct NODE{

int data;

struct NODE \*link;

};

typedef struct NODE node;

node \*new,\*front=NULL,\*rear=NULL,\*temp,\*temp1;

void insert();

void delete();

void display();

void main()

{

int choice;

while(1)

{

printf("1.Insert \n 2.Delete \n 3.Display \n 4.Exit \n");

printf("Enter the choice:");

scanf("%d",&choice);

switch(choice)

{

case 1: insert();

break;

case 2:delete();

break;

case 3: display();

break;

case 4: exit(0);

break;

default:printf("Wrong Choice");

}

}

getch();

}

void insert()

{

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

printf("enter an element to be inserted:");

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

if(front==NULL&&rear==NULL)

{

front=new;

rear=new;

front->link=NULL;

}

else

{

rear->link=new;

rear=new;

rear->link=NULL;

}

}

void delete()

{

if(front==NULL)

{

printf("queue is empty\n");

return;

}

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

temp=front;

front=front->link;

free(temp);

}

void display()

{

if(front==NULL)

{

printf("queue is empty\n");

return;

}

temp1=front;

while(temp1!=NULL)

{

printf("%d\n",temp1->data);

temp1=temp1->link;

}

}

Output:

1.Insert

2.Delete

3.Display

4.Exit

Enter the choice:1

enter an element to be inserted:5

1.Insert

2.Delete

3.Display

4.Exit

Enter the choice:1

enter an element to be inserted:2

1.Insert

2.Delete

3.Display

4.Exit

Enter the choice:1

enter an element to be inserted:7

1.Insert

2.Delete

3.Display

4.Exit

Enter the choice:3

5

2

7

1.Insert

2.Delete

3.Display

4.Exit

Enter the choice:2

deleted element is 5

1.Insert

2.Delete

3.Display

4.Exit

Enter the choice:3

2

7

1.Insert

2.Delete

3.Display

4.Exit