December 05, 2021

IMPLEMENTATION OF QUEUE USING LINK LIST

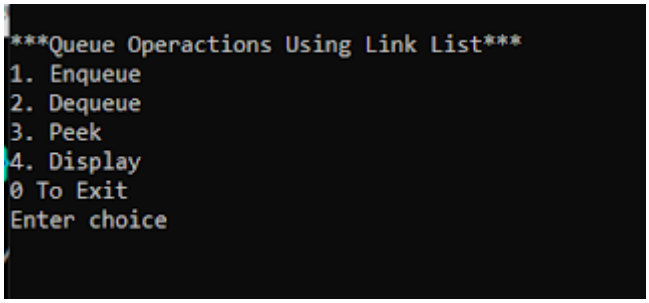
—

```c
/*
implemaintaction Of Queue using Link list
*/
#include<stdio.h>
#include<stdlib.h>

struct node{
        int data;
        struct node *next;
};

struct node *front=0;
struct node *rear=0;

//Function Declaraction
void Enqueue();
void Dequeue();
void Peek();
void Display();

//Driver program
int main(){
        int choice;
        do{
                printf("\n***Queue Operactions Using Link List***\n");
                printf("1. Enqueue\n");
                printf("2. Dequeue\n");
                printf("3. Peek\n");
```

```
29.                 printf("4. Display\n");
30.                 printf("0 To Exit\n");
31.                 printf("Enter choice ");
32.                 scanf("%d",&choice);
33.
34.                 switch(choice){
35.                         case 1:
36.                                 Enqueue();
37.                                 break;
38.                         case 2:
39.                                 Dequeue();
40.                                 break;
41.                         case 3:
42.                                 Peek();
43.                                 break;
44.                         case 4:
45.                                 Display();
46.                                 break;
47.                         default:
48.                                 printf("\n!Wrong Choice!\n");
49.                 }
50.         }while(choice!=0);
51.         return 0;
52. }
53.
54. //Enqueue Operaction
55. void Enqueue(){
56.         struct node *newnode;
57.         newnode=(struct node*)malloc(sizeof(struct node));
58.         printf("Enter Data ");
59.         scanf("%d",&newnode->data);
60.         newnode->next=0;
61.         if(front==0&&rear==0){
62.                 front=rear=newnode;
63.         }
64.         else{
65.                 rear->next=newnode;
66.                 rear=newnode;
67.         }
68. }
69.
70. //Dequeue Operaction
71. void Dequeue(){
```

```
72.         struct node *temp;
73.         if(front==0 && rear==0){
74.                 printf("\nQueue Is Empty\n");
75.         }
76.         else{
77.                 temp=front;
78.                 front=front->next;
79.                 free(temp);
80.                 printf("\n Deleted Successfully\n");
81.         }
82. }
83.
84. //Peek Operaction
85. void Peek(){
86.         if(front==0 && rear==0){
87.                 printf("\nQueue Is Empty\n");
88.         }
89.         else{
90.                 printf("\nFront Data Is %d",front->data);
91.         }
92. }
93.
94. //Displays The Queue
95. void Display(){
96.         struct node *temp;
97.         temp=front;
98.         if(front==0 && rear==0){
99.                 printf("\nQueue Is Empty\n");
100.        }
101.        else{
102.                printf("\n");
103.                while(temp!=0){
104.                        printf("%d ",temp->data);
105.                        temp=temp->next;
106.                }
107.        }
108. }
```

OUTPUT



Share

Labels: Queue implementation using link list

COMMENTS

Enter your comment...

POPULAR POSTS