



## CODERS HOME

Learn Html, Learn Css, Learn Java Script, Learn C language, Learn C++ Language, Best Ways To learn Coding At Home, Learn Coding At home, How To learn coding free, learn coing from scrach, Best websites to learn Coding free, Learn Python SQL Ruby Php etc coding languages at Home,

HOME

December 05, 2021

## IMPLEMENTATION OF CIRCULAR QUEUE USING LINK LIST

---

```
1. /*
2.  implemaintaction Of Circular Queue Using Link List
3. */
4. #include<stdio.h>
5. #include<stdlib.h>
6.
7. struct node{
8.     int data;
9.     struct node *next;
10. };
11.
12. struct node *front=0,*rear=0;
13.
14. void Enqueue();
15. void Dequeue();
16. void Peek();
17. void Display();
18.
19.
20. int main(){
21.     int choice;
22.     do{
23.         printf("\n***CIRCULAR QUEUE USING LINK LIST***\n");
24.         printf("1. ENQUEUE\n");
25.         printf("2. DEQUEUE\n");
26.         printf("3. PEEK\n");
27.         printf("4. DISPLAY\n");
28.         printf("0 TO EXIT\n");
```

```

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

```

```

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

```

**OUTPUT**

```
***CIRCULAR QUEUE USING LINK LIST***
1. ENQUEUE
2. DEQUEUE
3. PEEK
4. DISPLAY
0 TO EXIT
Enter Choice █
```

Share

Labels: implementation of Circular Queue using link list

#### COMMENTS



Enter your comment...

#### POPULAR POSTS