

College C codes > DSA-ASS-8 > C 8_1_priority_queue.c > enqueue(int, int)

```
1  /* Implement a priority queue using a linked list. The queue has the following basic operations:
2  • enqueue() - Insert an element in the queue.
3  • dequeue() - Remove the element from the queue.
4  • peek() - Return the element at the front node of the queue
5  • isfull() - Validates if the queue is full or not.
6  • isempty() - Checks if the queue is empty. */
7
8  #include <stdio.h>
9  #include <stdlib.h>
10
11 struct queue{
12     int data, pri;
13     struct queue *next, *prev;
14 };
15
16 struct queue *createnode(int x, int prior){
17     struct queue *newnode = (struct queue *)malloc(sizeof(struct queue));
18     newnode->data = x;
19     newnode->next = NULL;
20     newnode->prev = NULL;
21     newnode->pri = prior;
22     return newnode;
23 }
24
25 struct queue *front = NULL;
26 struct queue *rear = NULL;
27 int c = 0; // no. of queue elements
28
29 int isempty(){
30     if(rear == NULL){
31         return 1;
32     }
33     else{
34         return 0;
35     }
36 }
37
38 void enqueue(int x, int prior){
39     if(isempty()){
40         rear = createnode(x, prior);
41         front = rear;
42         c++;
43         return;
44     }
45     struct queue *newnode = createnode(x, prior);
46     if(prior < front->pri){ // priority of node is greater than priority of front node (priority increases in descending order)
47         front->next = newnode;
48         newnode->prev = front;
49         front = newnode;
```

```

50         c++;
51         return;
52     }
53     if(rear->pri <= newnode->pri){
54         rear->prev = newnode;
55         newnode->next = rear;
56         rear = newnode;
57         c++;
58         return;
59     }
60     struct queue *pivot = rear;
61     while(pivot->next != NULL && pivot->next->pri < pivot->pri) {
62         pivot = pivot->next;
63     }
64     newnode->next = pivot->next;
65     pivot->next->prev = newnode;
66     pivot->next = newnode;
67     newnode->prev = pivot;
68     c++;
69 }
70
71 void dequeue(){
72     if(isempty()){
73         printf("ERROR : Queue Underflow.");
74         return;
75     }
76     struct queue *temp = front;
77     front = front->prev;
78     front->next = NULL;
79     free(temp);
80     c--;
81 }
82
83 int peek(){
84     if(isempty()){
85         printf("ERROR : Queue is Empty");
86         return -1;
87     }
88     else{
89         return front->data;
90     }
91 }
92
93 int lenqueue(){
94     return c;
95 }
96

```

```

97 void display(){
98     struct queue *temp = rear;
99     printf("Queue (data,priority) : ");
100     while(temp != NULL){
101         printf("(%d, %d) ",temp->data,temp->pri);
102         temp = temp->next;
103     }
104     printf("\n");
105 }
106
107 void isfull(){
108     printf("Linked List implementation of queue is never full.\n");
109 }
110
111 int main(){
112     int ch;
113     while(1){
114         printf("\nEnter Your Choice :--\n1 - Insert an element in the queue\n2 - Remove Element from the queue\n3 - Display the element at the front of the queue\n4 - Check the
length of queue\n5 - Checks if the queue is empty\n6 - Display the queue\n7 - Check whether the queue is full or not\n0 - Exit.\nChoice : ");
115         scanf("%d",&ch);
116         switch(ch){
117             case 1 : int n, pri;
118                     printf("Enter the number (with priority) you want to insert : ");
119                     scanf("%d %d",&n,&pri);
120                     enqueue(n, pri);
121                     break;
122             case 2 : dequeue();
123                     break;
124             case 3 : printf("Element in the front of the queue : %d\n",peek());
125                     break;
126             case 4 : printf("Length of queue : %d\n",lenqueue());
127                     break;
128             case 5 : if(isempty()){
129                         printf("Queue is Empty\n");
130                     }
131                     else{
132                         printf("Queue is NOT empty\n");
133                     }
134                     break;
135             case 6 : display();
136                     break;
137             case 7 : isfull();
138                     break;
139             case 0 : exit(0);
140                     break;
141             default : printf("Wrong Input! Please Try Again.\n");
142         }
143     }
144     return 0;

```

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 1

Enter the number (with priority) you want to insert : 14 3

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 1

Enter the number (with priority) you want to insert : 10 1

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 1

Enter the number (with priority) you want to insert : 13 2

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 1

Enter the number (with priority) you want to insert : 15 3

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 1

Enter the number (with priority) you want to insert : 12 1

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 6

Queue (data,priority) : (15, 3) (14, 3) (13, 2) (12, 1) (10, 1)

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 2

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 6

Queue (data,priority) : (15, 3) (14, 3) (13, 2) (12, 1)

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 1

Enter the number (with priority) you want to insert : 69 2

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 6

Queue (data,priority) : (15, 3) (14, 3) (69, 2) (13, 2) (12, 1)

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 3

Element in the front of the queue : 12

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 4

Length of queue : 5

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 5

Queue is NOT empty

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 7

Linked List implementation of queue is never full.

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check whether the queue is full or not
- 0 - Exit.

Choice : 0

PS C:\Users\shuvr\OneDrive\Documents\CODING>

College C codes > DSA-ASS-8 > 8_2_circular_queue.c > display()

```
1  /* Implement a Circular queue using a linked list. The queue has the following basic operations:
2  • enqueue() - Insert an element in the queue.
3  • dequeue() - Remove the element from the queue.
4  • peek() - Return the element at the rear node of the queue
5  • isfull() - Validates if the queue is full or not.
6  • isempty() - Checks if the queue is empty. */
7
8  #include <stdio.h>
9  #include <stdlib.h>
10
11  struct queue{
12      int data;
13      struct queue *next, *prev;
14  };
15
16  int c=0;
17  struct queue *front = NULL;
18  struct queue *rear = NULL;
19
20  struct queue *createnode(int x){
21      struct queue *newnode = (struct queue *)malloc(sizeof(struct queue));
22      newnode->data = x;
23      newnode->next = NULL;
24      newnode->prev = NULL;
25      return newnode;
26  }
27
28  int isempty(){
29      if(rear == NULL){
30          return 1;
31      }
32      else{
33          return 0;
34      }
35  }
36
```

```

37 void enqueue(int x){
38     if(isempty()){
39         rear = createnode(x);
40         front = rear;
41         c++;
42         return;
43     }
44     struct queue *newnode = createnode(x);
45     newnode->next = rear;
46     rear->prev = newnode;
47     front->next = newnode;
48     newnode->prev = front;
49     rear = newnode;
50     c++;
51 }
52
53 void dequeue(){
54     if(isempty()){
55         printf("ERROR : Queue Underflow");
56         return;
57     }
58     struct queue *temp = front;
59     front = front->prev;
60     front->next = NULL;
61     rear->prev = front;
62     free(temp);
63     c--;
64 }
65
66 int peek(){
67     if(isempty()){
68         printf("ERROR : Queue is Empty");
69         return -1;
70     }
71     return front->data;
72 }
73
74 void isfull(){
75     printf("Linked List implementation of queue is never full\n");
76 }
77
78 int lenqueue(){
79     return c;
80 }
81

```

```

82 void display(){
83     if(isempty()){
84         printf("ERROR : Queue is Empty");
85         return;
86     }
87     printf("Queue : ");
88     struct queue *temp = rear;
89     for(int i=0; i < lenqueue(); i++){
90         printf(" %d <->", temp->data);
91         temp = temp->next;
92     }
93     printf("\n");
94 }
95
96 int main(){
97     int ch;
98     while(1){
99         printf("\nEnter Your Choice :--\n1 - Insert an element in the queue\n2 - Remove Element from the queue\n3 - Display the element at the front of the queue\n4 - Check the
100         length of queue\n5 - Checks if the queue is empty\n6 - Display the queue\n7 - Check if the queue is full or not\n8 - Exit.\nChoice : ");
101         scanf("%d",&ch);
102         switch(ch){
103             case 1 : int n;
104                     printf("Enter the number you want to insert : ");
105                     scanf("%d",&n);
106                     enqueue(n);
107                     break;
108             case 2 : dequeue();
109                     break;
110             case 3 : printf("Element in the front of the queue : %d\n",peek());
111                     break;
112             case 4 : printf("Length of queue : %d\n",lenqueue());
113                     break;
114             case 5 : if(isempty()){
115                         printf("Queue is Empty\n");
116                     }
117                     else{
118                         printf("Queue is NOT empty\n");
119                     }
120                     break;
121             case 6 : display();
122                     break;
123             case 7 : isfull();
124                     break;
125             case 8 : exit(0);
126                     break;
127             default : printf("Wrong Input! Please Try Again.\n");
128         }
129     }
130     return 0;

```

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check if the queue is full or not
- 0 - Exit.

Choice : 1

Enter the number you want to insert : 1

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check if the queue is full or not
- 0 - Exit.

Choice : 1

Enter the number you want to insert : 2

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check if the queue is full or not
- 0 - Exit.

Choice : 1

Enter the number you want to insert : 3

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check if the queue is full or not
- 0 - Exit.

Choice : 6

Queue : 3 <-> 2 <-> 1 <->

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check if the queue is full or not
- 0 - Exit.

Choice : 2

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check if the queue is full or not
- 0 - Exit.

Choice : 6

Queue : 3 <-> 2 <->

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check if the queue is full or not
- 0 - Exit.

Choice : 3

Element in the front of the queue : 2

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check if the queue is full or not
- 0 - Exit.

Choice : 4

Length of queue : 2

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check if the queue is full or not
- 0 - Exit.

Choice : 5

Queue is NOT empty

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check if the queue is full or not
- 0 - Exit.

Choice : 7

Linked List implementation of queue is never full

Enter Your Choice :--

- 1 - Insert an element in the queue
- 2 - Remove Element from the queue
- 3 - Display the element at the front of the queue
- 4 - Check the length of queue
- 5 - Checks if the queue is empty
- 6 - Display the queue
- 7 - Check if the queue is full or not
- 0 - Exit.

Choice : 0

PS C:\Users\shuvr\OneDrive\Documents\CODING>