

# Harsh\_Kumar\_MAN\_106\_Assignment\_4

- Answer 1:

```
1  #include<bits/stdc++.h>
2  using namespace std;
3
4  struct node{
5      int element;
6      node *next;
7  };
8
9  class QUEUE{
10     node *q_front;
11     node *q_rear;
12     int size;
13 public:
14     QUEUE();
15     ~QUEUE();
16     bool underflow();
17     void add(int item);
18     int remove();
19     int count();
20 };
21
22 QUEUE::QUEUE(){
23     q_front = NULL;
24     q_rear = NULL;
25     size = 0;
26 }
27
28 QUEUE::~~QUEUE(){
29     while (!underflow()) remove();
30     delete q_front, q_rear;
31 }
32
33 bool QUEUE::underflow(){
34     if (q_front==NULL) return true;
35     else return false;
36 }
```

```

38 void QUEUE::add(int item){
39     node *next_element;
40     next_element = new node;
41     next_element->element = item;
42     next_element->next = NULL;
43     if (q_rear == NULL) q_front = next_element;
44     else q_rear->next = next_element;
45     size++;
46     q_rear = next_element;
47 }
48
49 int QUEUE::remove(){
50     if (underflow()){
51         cout << "Queue is empty. Add some values first\n";
52         return -1;
53     }
54     else{
55         node *temp;
56         temp = q_front;
57         int item = temp->element;
58         q_front = temp->next;
59         size--;
60         delete temp;
61         return item;
62     }
63 }
64
65 int QUEUE::count(){
66     return size;
67 }

```

```

69✓ int main(){
70     QUEUE q;
71     for (int i=0; i<10; i++) q.add(i+1);
72     cout << "There are " << q.count() << " elements in the queue\n";
73✓ for (int i=0; i<11; i++){
74     cout << q.remove() << '\n';
75     cout << "There are " << q.count() << " elements in the queue\n";
76 }
77 }
78

```

- Output :

```
There are 10 elements in the queue
1
There are 9 elements in the queue
2
There are 8 elements in the queue
3
There are 7 elements in the queue
4
There are 6 elements in the queue
5
There are 5 elements in the queue
6
There are 4 elements in the queue
7
There are 3 elements in the queue
8
There are 2 elements in the queue
9
There are 1 elements in the queue
10
There are 0 elements in the queue
Queue is empty. Add some values first
-1
There are 0 elements in the queue
```

- Answer 2\_1 :

```

1  #include<bits/stdc++.h>
2  using namespace std;
3
4  struct node{
5      char element;
6      node *next;
7  };
8
9  class QUEUE{
10     node *q_front;
11     node *q_rear;
12     int size;
13 public:
14     QUEUE();
15     ~QUEUE();
16     bool underflow();
17     void add(char item);
18     char remove();
19     int count();
20 };
21
22 QUEUE::QUEUE(){
23     q_front = NULL;
24     q_rear = NULL;
25     size = 0;
26 }
27
28 QUEUE::~~QUEUE(){
29     while (!underflow()) remove();
30     delete q_front, q_rear;
31 }
32
33 bool QUEUE::underflow(){
34     if (q_front==NULL) return true;
35     else return false;
36 }

```

```

38 void QUEUE::add(char item){
39     node *next_element;
40     next_element = new node;
41     next_element->element = item;
42     next_element->next = NULL;
43     if (q_rear == NULL) q_front = next_element;
44     else q_rear->next = next_element;
45     size++;
46     q_rear = next_element;
47 }
48
49 char QUEUE::remove(){
50     if (underflow()){
51         cout << "Queue is empty. Add some values first\n";
52         return '-';
53     }
54     else{
55         node *temp;
56         temp = q_front;
57         char item = temp->element;
58         q_front = temp->next;
59         size--;
60         delete temp;
61         return item;
62     }
63 }
64
65 int QUEUE::count(){
66     return size;
67 }
68

```

```

69 int main(){
70     QUEUE q;
71     for (int i=65; i<75; i++) q.add(char(i));
72     cout << "There are " << q.count() << " elements in the queue\n";
73     for (int i=0; i<11; i++){
74         cout << q.remove() << '\n';
75         cout << "There are " << q.count() << " elements in the queue\n";
76     }
77 }

```

- Output :

```
.....
There are 10 elements in the queue
A
.....
There are 9 elements in the queue
B
.....
There are 8 elements in the queue
C
.....
There are 7 elements in the queue
D
.....
There are 6 elements in the queue
E
.....
There are 5 elements in the queue
F
.....
There are 4 elements in the queue
G
.....
There are 3 elements in the queue
H
.....
There are 2 elements in the queue
I
.....
There are 1 elements in the queue
J
.....
There are 0 elements in the queue
Queue is empty. Add some values first
-
.....
There are 0 elements in the queue
```

- **Answer 2\_2:**

```

1  #include<bits/stdc++.h>
2  using namespace std;
3
4  struct node{
5      string element;
6      node *next;
7  };
8
9  class QUEUE{
10     node *q_front;
11     node *q_rear;
12     int size;
13 public:
14     QUEUE();
15     ~QUEUE();
16     bool underflow();
17     void add(string item);
18     string remove();
19     int count();
20 };
21
22 QUEUE::QUEUE(){
23     q_front = NULL;
24     q_rear = NULL;
25     size = 0;
26 }
27
28 QUEUE::~~QUEUE(){
29     while (!underflow()) remove();
30     delete q_front, q_rear;
31 }
32
33 bool QUEUE::underflow(){
34     if (q_front==NULL) return true;
35     else return false;
36 }

```

```

38 void QUEUE::add(string item){
39     node *next_element;
40     next_element = new node;
41     next_element->element = item;
42     next_element->next = NULL;
43     if (q_rear == NULL) q_front = next_element;
44     else q_rear->next = next_element;
45     size++;
46     q_rear = next_element;
47 }
48
49 string QUEUE::remove(){
50     if (underflow()){
51         cout << "Queue is empty. Add some values first\n";
52         return "NA";
53     }
54     else{
55         node *temp;
56         temp = q_front;
57         string item = temp->element;
58         q_front = temp->next;
59         size--;
60         delete temp;
61         return item;
62     }
63 }
64
65 int QUEUE::count(){
66     return size;
67 }
}

69 int main(){
70     QUEUE q;
71     string s = "Hello World";
72     for (int i=65; i<70; i++){
73         s.push_back(i);
74         q.add(s);
75     }
76     cout << "There are " << q.count() << " elements in the queue\n";
77     for (int i=0; i<6; i++){
78         cout << q.remove() << '\n';
79         cout << "There are " << q.count() << " elements in the queue\n";
80     }
81 }
82

```

- Output :



```

There are 5 elements in the queue
Hello WorldA
There are 4 elements in the queue
Hello WorldAB
There are 3 elements in the queue
Hello WorldABC
There are 2 elements in the queue
Hello WorldABCD
There are 1 elements in the queue
Hello WorldABCDE
There are 0 elements in the queue
Queue is empty. Add some values first
NA
There are 0 elements in the queue

```

- Answer 3:

```

1  #include<bits/stdc++.h>
2  using namespace std;
3
4  struct node{
5      int element;
6      node *next;
7  };
8
9  class C_QUEUE{
10     node *q_front;
11     node *q_rear;
12     int size;
13 public:
14     C_QUEUE();
15     ~C_QUEUE();
16     bool underflow();
17     void add(int item);
18     int remove();
19     int count();
20 };
21
22 C_QUEUE::C_QUEUE(){
23     q_front = NULL;
24     q_rear = NULL;
25     size = 0;
26 }
27
28 C_QUEUE::~~C_QUEUE(){
29     while (!underflow()) remove();
30 }
31
32 bool C_QUEUE::underflow(){
33     if (q_front==NULL) return true;
34     else return false;

```

```

35 }
36
37 void C_QUEUE::add(int item){
38     node *next_element;
39     next_element = new node;
40     next_element->element = item;
41     next_element->next = q_front;
42     if (q_rear == NULL) q_front = next_element;
43     else q_rear->next = next_element;
44     size++;
45     q_rear = next_element;
46 }
47
48 int C_QUEUE::remove(){
49     if (underflow()){
50         cout << "Queue is empty. Add some values first\n";
51         return -1;
52     }
53     else if (q_front==q_rear){
54         int item = q_front->element;
55         delete(q_rear);
56         q_front=NULL;
57         size--;
58         return item;
59     }
60     else{
61         node *temp;
62         temp = q_front;
63         int item = temp->element;
64         q_front = temp->next;
65         size--;
66         delete temp;
67         return item;
68     }
69 }
70
71 int C_QUEUE::count(){
72     return size;
73 }
74
75 int main(){
76     C_QUEUE q;
77     for (int i=0; i<10; i++) q.add(i+1);
78     cout << "There are " << q.count() << " elements in the queue\n";
79     for (int i=0; i<11; i++){
80         cout << q.remove() << '\n';
81         cout << "There are " << q.count() << " elements in the queue\n";
82     }
83 }
84

```

Output :

```

There are 10 elements in the queue
1
There are 9 elements in the queue
2
There are 8 elements in the queue
3
There are 7 elements in the queue
4
There are 6 elements in the queue
5
There are 5 elements in the queue
6
There are 4 elements in the queue
7
There are 3 elements in the queue
8
There are 2 elements in the queue
9
There are 1 elements in the queue
10
There are 0 elements in the queue
Queue is empty. Add some values first
-1
There are 0 elements in the queue

```

- **Answer 4:**

```

1  #include<bits/stdc++.h>
2  using namespace std;
3
4  struct node{
5      float element;
6      node *next;
7  };
8
9  class STACK{
10     node *top;
11 public:
12     STACK();
13     ~STACK();
14     bool underflow();
15     void push(float item);
16     float pop();
17 };
18
19 STACK::STACK(){
20     top = NULL;
21 }
22
23 STACK::~~STACK(){
24     while (!underflow()) pop();
25     delete top;
26 }
27
28 bool STACK::underflow(){
29     if (top==NULL) return true;
30     else return false;
31 }

```

```

33✓ void STACK::push(float item){
34     node *next_element;
35     next_element = new node;
36     next_element->element = item;
37     next_element->next = top;
38     top = next_element;
39 }
40
41✓ float STACK::pop(){
42✓     if (underflow()){
43         cout << "Stack underflow. Push some values first\n";
44         return -1;
45     }
46✓     else{
47         node *temp;
48         temp = top;
49         float item = temp->element;
50         top = temp->next;
51         delete temp;
52         return item;
53     }
54 }
55
56✓ int main(){
57     STACK s, t;
58     for (int i=0; i<10; i++) s.push(float(i+1)+0.12);
59     cout << "Printing initial stack and inverting it:\n";
60✓     for (int i=0; i<10; i++) {
61         float last_element = s.pop();
62         cout << last_element << '\n';
63         t.push(last_element);
64     }
65     cout << "Printing inverted stack:\n";
66     for (int i=0; i<10; i++) cout << t.pop() << '\n';
67 }

```

- Output :

```

Printing initial stack and inverting it:
10.12
9.12
8.12
7.12
6.12
5.12
4.12
3.12
2.12
1.12
Printing inverted stack:
1.12
2.12
3.12
4.12
5.12
6.12
7.12
8.12
9.12
10.12

```