



Queue

Queue

A queue is a linear list of elements in which deletions can take place only at one end called the front, and insertions can take place only at the end called the rear. The queue is a First In First Out type of data structure (**FIFO**), the terms **FRONT** and **REAR** are used in describing a linear list only when it is implemented as a queue.

Queue in [STL](#).

Syntax:

```
queue<object_type> variable_name;
```

Example:

```
queue<int> q;  
queue<string> q;
```

My Code:

```

// Containers --> Queue (FIFO --> First in first out)
#include <bits/stdc++.h>
using namespace std;
void explainQueue()
{
    queue<int> q;
    q.push(1); //{1}
    q.push(2); //{1,2}
    q.push(4); //{1,2,4}

    q.back() += 5; // add 5 in the last element (4+5=9)

    cout << q.back(); // print 9

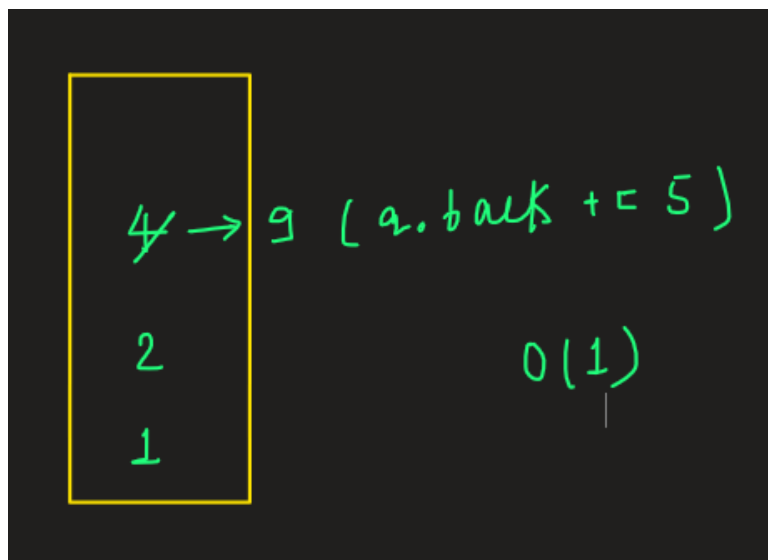
    // Q is {1,2,9}
    cout << q.front(); //print 1

    q.pop(); // {2,9}

    cout << q.front(); // print 2

    // size, swap, empty same as stack
}
int main()
{
    explainQueue();
    return 0;
}

```



Functions in Queue:

push() – to insert an element in the queue.

```
queue<int> q  
q.push(110);  
q.push(220);
```

pop() – deletes the first element of the queue.

```
q.pop();
```

front() – returns a reference to the first element of the queue.

```
q.front();
```

back() – returns a reference to the last element of the queue.

```
q.back();
```

emplace() – to insert an element in the queue.

```
queue<int> q;  
q.emplace(1);  
q.emplace(2);
```

size() – returns the number of elements on the queue.

```
q.size();
```

empty() – to check if the queue is empty or not.

```
q.empty();
```

Striver's code:

```
#include<bits/stdc++.h>
using namespace std;
void printqueue(queue<int> q1)
{
    queue<int> q2=q1;
    while(!q2.empty())
    {
        cout<<q2.front()<<"\\n";
        q2.pop();
    }
}
int main()
{
    queue<int> q;
    for(int i=1;i<=5;i++)
        q.push(i);

    cout<<"The elements of the queue are:"<<endl;
    printqueue(q);

    cout<<"The size of the queue: "<<q.size()<<endl;
    cout<<"The front element of the queue: "<<q.front()<<endl;
    cout<<"The last element of the queue: "<<q.back()<<endl;
    cout<<"Pop the front element: "<<endl;
    q.pop();
    printqueue(q);
}
```

Output:

```
The elements of the queue are:
1
2
3
4
5
The size of the queue: 5
The front element of the queue: 1
The last element of the queue: 5
Pop the front element:
2
3
4
5
```