

QUEUE

AIM:

To create a user defined queue by using the list in C++.

ALGORITHM:

- Create a queue using a list.
- Display the menu with queue operations.
- Read the user's choice.
- If choice is 1, check whether the queue is empty and display the result.
- If choice is 2, display the size of the queue.
- If choice is 3, display the front element of the queue.
- If choice is 4, display the back element of the queue.
- If choice is 5, read a value from the user and insert it into the queue.
- If choice is 6, remove the front element from the queue.
- If choice is 7, display all elements of the queue.
- Repeat the process until the user selects exit.

PROGRAM:

```
/*
 * Program to demonstrate queue using the list
 * Author : MUTHUGANESH S
 * Date   : 06/2/2026
 * Filename: Queue.cpp
 * retval  : void
 */

#include <iostream>
#include <list>
using namespace std;

class Queue {
    list<int> q;

public:
    bool empty() {
```

```

        return q.empty();
    }

    int size() {
        return q.size();
    }

    void swap(){
        q.swap(q);
    }

    int front() {
        if (!q.empty())
            return q.front();
    }

    int back() {
        if (!q.empty())
            return q.back();
    }

    void push(int value) {
        q.push_back(value);
    }

    void pop() {
        if (!q.empty())
            q.pop_front();
    }

    void display() {
        for (int value : q) {
            cout << value << " ";
        }
        cout << endl;
    }
};

int main(){
    Queue q;

    cout<<"1.check empty\n2.size\n3.front\n4.back\n5.push\n";
    cout<<"6.pop\n7.display\n8.exit\n";
    int choice, value;

    while(choice!=8){

        cout<<"\nEnter your choice: ";
        cin>>choice;

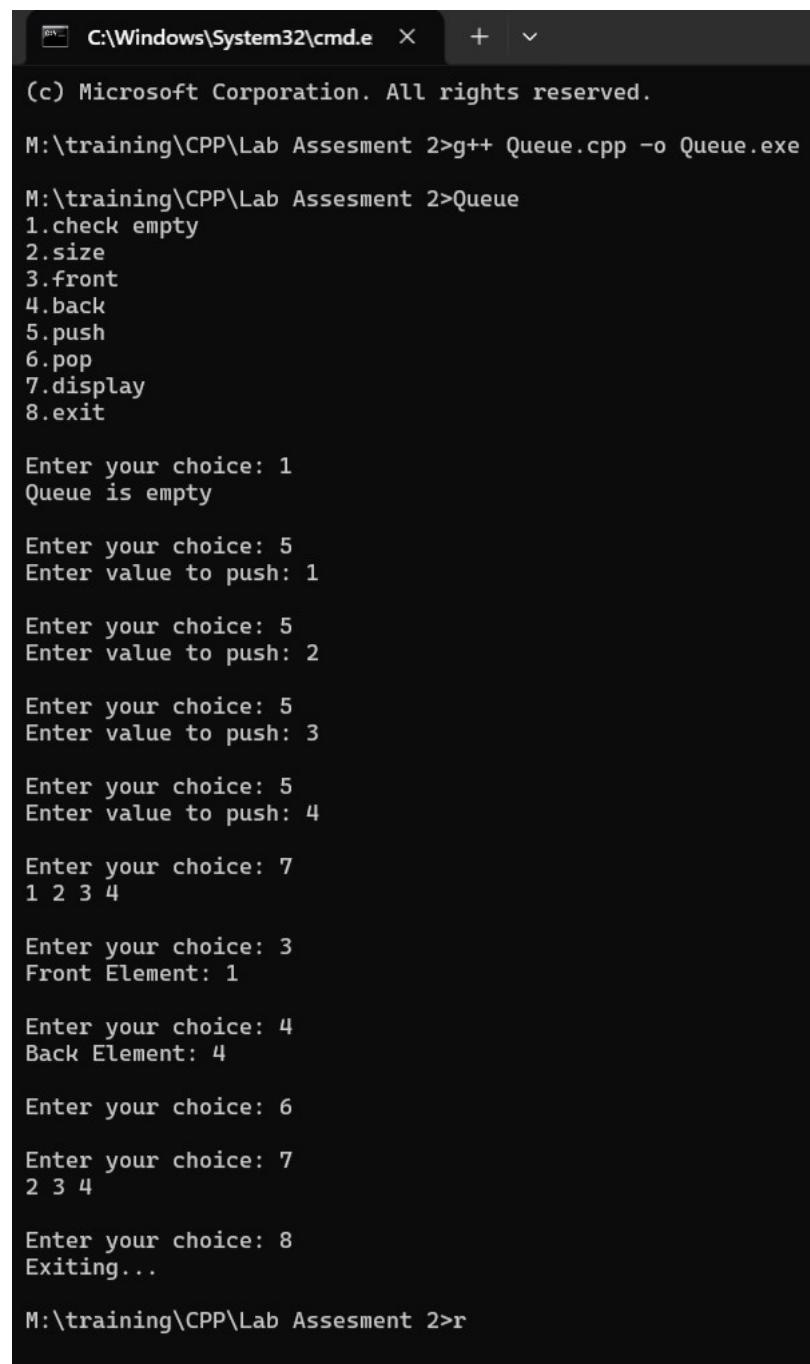
```

```
switch(choice){  
  
    // Check if queue is empty  
    case 1:  
  
        if(q.empty())  
            cout<<"Queue is empty"<<endl;  
        else  
            cout<<"Queue is not empty"<<endl;  
        break;  
  
    // Size of the queue  
    case 2:  
        cout<<"Size of Queue: "<<q.size()<<endl;  
        break;  
  
    // Accessing next elements  
    case 3:  
        cout<<"Front Element: "<<q.front()<<endl;  
        break;  
  
    // Accessing last elements  
    case 4:  
        cout<<"Back Element: "<<q.back()<<endl;  
        break;  
  
    // Adding elements to the queue  
    case 5:  
        cout<<"Enter value to push: ";  
        cin>>value;  
        q.push(value);  
        break;  
  
    // Removing elements from the queue  
    case 6:  
        q.pop();  
        break;  
  
    // Displaying the queue  
    case 7:  
        q.display();  
        break;  
  
    // Exiting the program  
    case 8:  
        cout << "Exiting..." << endl;  
        break;
```

```
        default:
            cout << "Invalid choice! Please try again." <<
endl;
    }
}

return 0;
}
```

OUTPUT:



The screenshot shows a Windows Command Prompt window titled 'C:\Windows\System32\cmd.exe'. The window displays the following interaction with a C++ program:

```
(c) Microsoft Corporation. All rights reserved.

M:\training\CPP\Lab Assesment 2>g++ Queue.cpp -o Queue.exe

M:\training\CPP\Lab Assesment 2>Queue
1.check empty
2.size
3.front
4.back
5.push
6.pop
7.display
8.exit

Enter your choice: 1
Queue is empty

Enter your choice: 5
Enter value to push: 1

Enter your choice: 5
Enter value to push: 2

Enter your choice: 5
Enter value to push: 3

Enter your choice: 5
Enter value to push: 4

Enter your choice: 7
1 2 3 4

Enter your choice: 3
Front Element: 1

Enter your choice: 4
Back Element: 4

Enter your choice: 6

Enter your choice: 7
2 3 4

Enter your choice: 8
Exiting...

M:\training\CPP\Lab Assesment 2>r
```