QUEUE

AIM:

To create a Queue and perform the queue operations in C++.

ALGORITHM:

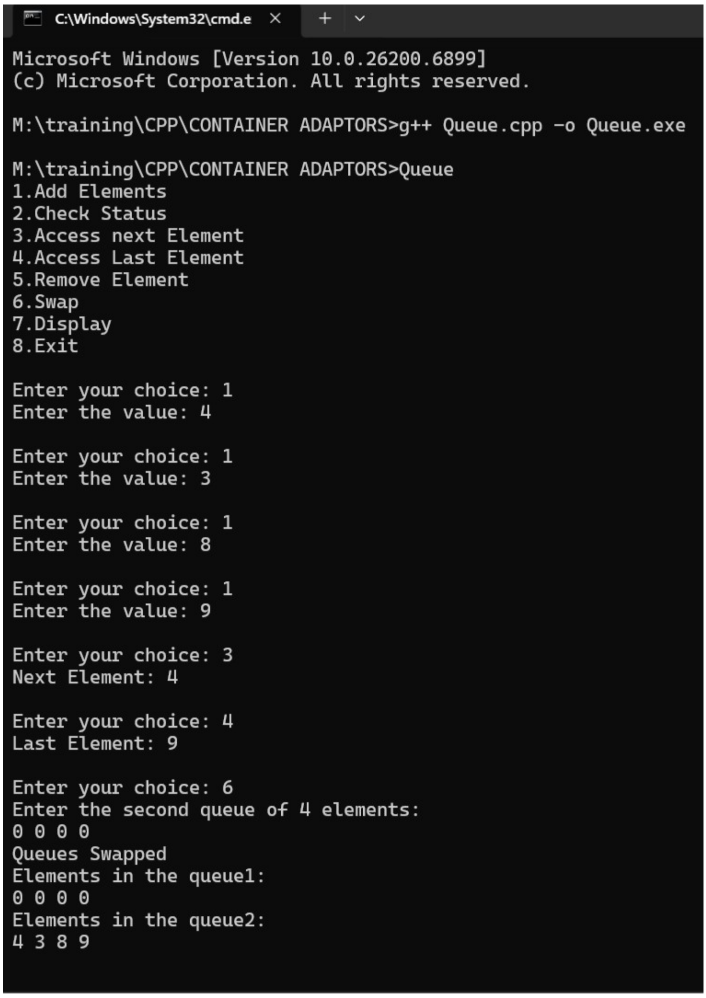
- Declare two integer queues q and q2 using STL queue.
- Display the menu of queue operations to the user.
- Repeat the menu until the user selects the exit option.
- Read the user’s choice.
- If choice is Add Element, read a value and insert it into the queue using push().
- If choice is Check Status, verify whether the queue is empty using empty() and display size if not empty.
- If choice is Access Next Element, display the front element using front().
- If choice is Access Last Element, display the rear element using back().
- If choice is Remove Element, delete the front element using pop().
- If choice is Swap Queues, read elements into second queue and exchange contents using swap().
- If choice is Display, traverse and display queue elements by passing a copy of the queue.
- If choice is Exit, terminate the loop.

PROGRAM:

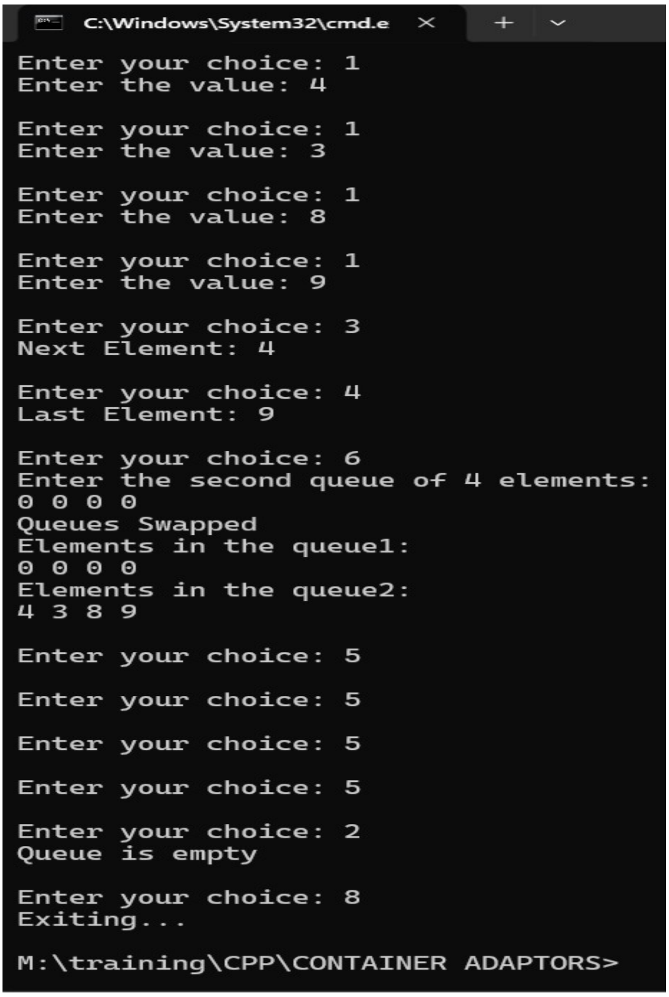
```cpp
/*
*  Program to demonstarte queue
*  Author  : MUTHUGANESH S
*  Date    : 21/1/2026
*  Filename: Queue.cpp
*  retval  : void
*/

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

// Function to display elements of the queue
void Display(queue <int> q){
    while(!q.empty()){
        cout<<q.front()<<" ";
        q.pop();
    }
    cout<<endl;
}
```

```
int main(void){

    queue <int> q;

    cout<<"1.Add Elements\n2.Check Status\n3.Access next Element\n";
    cout<<"4.Access Last Element\n5.Remove 
Element\n6.Swap\n7.Display\n8.Exit\n";

    int choice, value;
    queue <int> q2;

    while(choice!=8){

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

        switch(choice){

            // Adding elements to the queue
            case 1:
                cout<<"Enter the value: ";
                cin>>value;
                q.push(value);
                break;

            // Checking status
            case 2:
                if(q.empty())
                    cout<<"Queue is empty"<<endl;
                else
                    cout<<"Size of Queue: "<<q.size()<<endl;
                break;

            // Accessing next elements
            case 3:
                cout<<"Next Element: "<<q.front()<<endl;
                break;

            // Accessing last elements
            case 4:
                cout<<"Last Element: "<<q.back()<<endl;
                break;

            // Removing elements
            case 5:
                q.pop();
                break;

            // Swapping queues
            case 6:
                cout<<"Enter the second queue of "<<q.size()<<" elements:\n”;
                for(int i=0;i<q.size();i++){
                    cin>>value;
                    q2.push(value);
                }
                q.swap(q2);
                cout<<"Queues Swapped"<<endl;
                cout<<"Elements in the queue1: "<<endl;
                Display(q);
                cout<<"Elements in the queue2: "<<endl;
                Display(q2);
                break;

            // Displaying elements
            case 7:
                cout<<"Elements in the queue: "<<endl;
                Display(q);
                break;

            // Exiting
            case 8:
                cout<<"Exiting..."<<endl;
                break;
        }
    }
    return 0;
}
```

OUTPUT:



```
Microsoft Windows [Version 10.0.26200.6899]
(c) Microsoft Corporation. All rights reserved.

M:\training\CPP\CONTAINER ADAPTORS>g++ Queue.cpp -o Queue.exe

M:\training\CPP\CONTAINER ADAPTORS>Queue
1.Add Elements
2.Check Status
3.Access next Element
4.Access Last Element
5.Remove Element
6.Swap
7.Display
8.Exit

Enter your choice: 1
Enter the value: 4

Enter your choice: 1
Enter the value: 3

Enter your choice: 1
Enter the value: 8

Enter your choice: 1
Enter the value: 9

Enter your choice: 3
Next Element: 4

Enter your choice: 4
Last Element: 9

Enter your choice: 6
Enter the second queue of 4 elements:
0 0 0 0
Queues Swapped
Elements in the queue1:
0 0 0 0
Elements in the queue2:
4 3 8 9
```



```
Enter your choice: 1
Enter the value: 4

Enter your choice: 1
Enter the value: 3

Enter your choice: 1
Enter the value: 8

Enter your choice: 1
Enter the value: 9

Enter your choice: 3
Next Element: 4

Enter your choice: 4
Last Element: 9

Enter your choice: 6
Enter the second queue of 4 elements:
0 0 0 0
Queues Swapped
Elements in the queue1:
0 0 0 0
Elements in the queue2:
4 3 8 9

Enter your choice: 5

Enter your choice: 5

Enter your choice: 5

Enter your choice: 5

Enter your choice: 2
Queue is empty

Enter your choice: 8
Exiting...

M:\training\CPP\CONTAINER ADAPTORS>
```