Villas, Kaith Angel D.
CPE21S2

15/12/24
Sir Dennis Nava

Write a C++ or Python program that will implement one of the data structures such as an array, linked list, queue, stack, tree, or graph. The program should allow the user to perform the add, search, and delete operations.

Upload the source code file, screenshot of codes, and output.

CODE AND OUTPUT

[*] queue.cpp

```cpp
#include <iostream>
#include <queue>
#include <string>
#include<limits>
using namespace std;

void show(queue<string> ch) {
    if (ch.empty()) {
        cout << "The queue is empty." << endl;
        return;
    }
    queue<string> c = ch;
    cout << "Queue elements:\n";
    while (!c.empty()) {
        cout << c.front() << endl;
        c.pop();
    }
    cout << endl;
}


bool search(queue<string> ch, const string& target) {
    queue<string> c = ch;
    while (!c.empty()) {
        if (c.front() == target) {
            return true;
        }
        c.pop();
    }
    return false;
}

int main() {
    queue<string> charac;
    int choice;

    while (true) {
        cout << "\nMenu:\n";
        cout << "1. Add Element\n";
        cout << "2. Search Element\n";
        cout << "3. Delete Front Element\n";
        cout << "4. Show All Elements\n";
        cout << "5. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;
        cin.clear();
        cin.ignore(numeric_limits<int>::max(), '\n');
```

```
        switch (choice) {
            case 1: {
                string input;
                cout << "Enter an element to add: ";
                getline(cin, input);
                charac.push(input);
                cout << "Element added successfully.\n";
                break;
            }

            case 2: {
                string target;
                cout << "Enter the element to search: ";
                getline(cin, target);
                if (search(charac, target)) {
                    cout << "Element found in the queue.\n";
                } else {
                    cout << "Element not found in the queue.\n";
                }
                break;
            }

            case 3: {
                if (!charac.empty()) {
                    cout << "Deleted front element: " << charac.front() << endl;
                    charac.pop();
                } else {
                    cout << "The queue is already empty.\n";
                }
                break;
            }

            case 4: {
                show(charac);
                break;
            }

            case 5: {
                return 0;
            }

            default: {
                cout << "Invalid input!\n";
                break;
            }
        }
    }

    return 0;
}
```

```
Menu:
1. Add Element
2. Search Element
3. Delete Front Element
4. Show All Elements
5. Exit
Enter your choice: abc
Invalid input!

Menu:
1. Add Element
2. Search Element
3. Delete Front Element
4. Show All Elements
5. Exit
Enter your choice: 6
Invalid input!

Menu:
1. Add Element
2. Search Element
3. Delete Front Element
4. Show All Elements
5. Exit
Enter your choice: 1
Enter an element to add: ellie
Element added successfully.
```

```
Menu:
1. Add Element
2. Search Element
3. Delete Front Element
4. Show All Elements
5. Exit
Enter your choice: 1
Enter an element to add: dina
Element added successfully.

Menu:
1. Add Element
2. Search Element
3. Delete Front Element
4. Show All Elements
5. Exit
Enter your choice: 1
Enter an element to add: joel
Element added successfully.

Menu:
1. Add Element
2. Search Element
3. Delete Front Element
4. Show All Elements
5. Exit
Enter your choice: 2
Enter the element to search: dina
Element found in the queue.

Menu:
1. Add Element
2. Search Element
3. Delete Front Element
4. Show All Elements
5. Exit
Enter your choice: 4
Queue elements:
ellie
dina
joel
```

```
Menu:
1. Add Element
2. Search Element
3. Delete Front Element
4. Show All Elements
5. Exit
Enter your choice: 5

--------------------------------
Process exited after 51.86 seconds with return value 0
Press any key to continue . . .
```