DAY 4

GAURAV KUMAR 22BCS10159 KPIT-901(A)

Given a string s, find the first non-repeating character in it and return its index. If it does not exist, return -1.

```cpp
#include <iostream>
#include <unordered_map>
using namespace std;

int firstUniqChar(string s) {
    unordered_map<char, int> freq;

    for (char c : s) {
        freq[c]++;
    }

    for (int i = 0; i < s.size(); i++) {
        if (freq[s[i]] == 1) {
            return i;
        }
    }

    return -1;
}

int main() {
    string s;
    cout << "Enter the string: ";
    cin >> s;
    int result = firstUniqChar(s);
    if (result != -1) {
        cout << "The index of the first non-repeating character is: " << result << endl;
    } else {
        cout << "No non-repeating character found." << endl;
    }
    return 0;
}

```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ day4ques6.cpp -o day4ques6 && "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\"day4ques6
Enter the string: asdfgh
The index of the first non-repeating character is: 0

c:\Users\Gaurav Kumar\OneDrive\Desktop\class>
```

Design a stack that supports push, pop, top, and retrieving the minimum element in constant time.

day4ques1.cpp > main()

```cpp
#include <iostream>
#include <stack>
using namespace std;

class MinStack {
private:
    stack<int> mainStack;
    stack<int> minStack;

public:
    void push(int x) {
        mainStack.push(x);
        if (minStack.empty() || x <= minStack.top()) {
            minStack.push(x);
        }
    }

    void pop() {
        if (mainStack.top() == minStack.top()) {
            minStack.pop();
        }
        mainStack.pop();
    }

    int top() {
        return mainStack.top();
    }
    int getMin() {
        return minStack.top();
    }
};

int main() {
    MinStack stack;
    stack.push(5);
    stack.push(2);
    stack.push(8);
    stack.push(17);
    stack.push(10);
    stack.push(18);
    stack.push(9);
    stack.push(11);
    stack.push(19);

    cout << "Minimum: " << stack.getMin() << endl;
    stack.pop();
    cout << "Minimum: " << stack.getMin() << endl;
    stack.pop();
    cout << "Top: " << stack.top() << endl;
    cout << "Minimum: " << stack.getMin() << endl;

    return 0;
}
```

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ day4ques1.cpp -o day4ques1 && "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\"day4ques1
Minimum: 2
Minimum: 2
Top: 9
Minimum: 2
```

The school cafeteria offers circular and square sandwiches at lunch break, referred to by numbers 0 and 1 respectively. All students stand in a queue. Each student either prefers square or circular sandwiches.

day4ques2.cpp > main()

```cpp
#include <iostream>
#include <queue>
#include <vector>

using namespace std;

int countStudents(vector<int>& students, vector<int>& sandwiches) {
    queue<int> studentQueue;
    for (int student : students) {
        studentQueue.push(student);
    }
    int sandwichIndex = 0, attempts = 0;
    while (!studentQueue.empty() && attempts < studentQueue.size()) {
        if (studentQueue.front() == sandwiches[sandwichIndex]) {
            studentQueue.pop();
            sandwichIndex++;
            attempts = 0;
        } else {
            int temp = studentQueue.front();
            studentQueue.pop();
            studentQueue.push(temp);
            attempts++;
        }
    }
    return studentQueue.size();
}

int main() {
    int n;
    cout << "Enter the number of students: ";
    cin >> n;
    vector<int> students(n), sandwiches(n);
    cout << "Enter the preferences of students (0 for circular, 1 for square): ";
    for (int i = 0; i < n; ++i) {
        cin >> students[i];
    }
    cout << "Enter the sandwich stack (top to bottom: 0 for circular, 1 for square): ";
    for (int i = 0; i < n; ++i) {
        cin >> sandwiches[i];
    }
    int result = countStudents(students, sandwiches);
    cout << "Number of students unable to eat: " << result << endl;

    return 0;
}
```

Chat Add Logs Code Agent Improve Code Share Code Link

```
C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ day4ques2.cpp -o day4ques2 && "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\"day4ques2
Enter the number of students: 5
Enter the preferences of students (0 for circular, 1 for square): 1 0 1 0 1
Enter the sandwich stack (top to bottom: 0 for circular, 1 for square): 0 1 0 1 0
Number of students unable to eat: 1
```

Reverse queue

day4ques3.cpp > ...

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

void reverseQueue(queue<int>& q) {
    if (q.empty()) return;
    int front = q.front();
    q.pop();
    reverseQueue(q);
    q.push(front);
}

int main() {
    queue<int> q;
    int n, x;

    cout << "Enter the number of elements in the queue: ";
    cin >> n;
    cout << "Enter the elements of the queue: ";
    for (int i = 0; i < n; ++i) {
        cin >> x;
        q.push(x);
    }

    reverseQueue(q);

    cout << "Reversed queue: ";
    while (!q.empty()) {
        cout << q.front() << " ";
        q.pop();
    }

    return 0;
}
```

```
Active code page: 65001

C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ day4ques3.cpp -o day4ques3 && "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\"day4ques3
Enter the number of elements in the queue: 5
Enter the elements of the queue: 1 2 3 4 5
Reversed queue: 5 4 3 2 1
c:\Users\Gaurav Kumar\OneDrive\Desktop\class>
```

Balanced parentheses string

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

int scoreOfParentheses(string s) {
    stack<int> st;
    st.push(0);

    for (char c : s) {
        if (c == '(') {
            st.push(0);
        } else {
            int top = st.top();
            st.pop();
            int score = max(2 * top, 1);
            st.top() += score;
        }
    }

    return st.top();
}

int main() {
    string s;
    cout << "Enter a balanced parentheses string: ";
    cin >> s;

    int result = scoreOfParentheses(s);
    cout << "Score of the string: " << result << endl;

    return 0;
}
```

```
Active code page: 65001

C:\Users\Gaurav Kumar\OneDrive\Desktop\class>cd "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\" && g++ day4ques4.cpp -o day4ques4 && "c:\Users\Gaurav Kumar\OneDrive\Desktop\class\"day4ques4
Enter a balanced parentheses string: ()()(((()))()()(())
Score of the string: 11

c:\Users\Gaurav Kumar\OneDrive\Desktop\class>
```