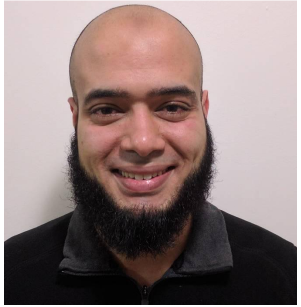
C++ Programming

STL Homework 1

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

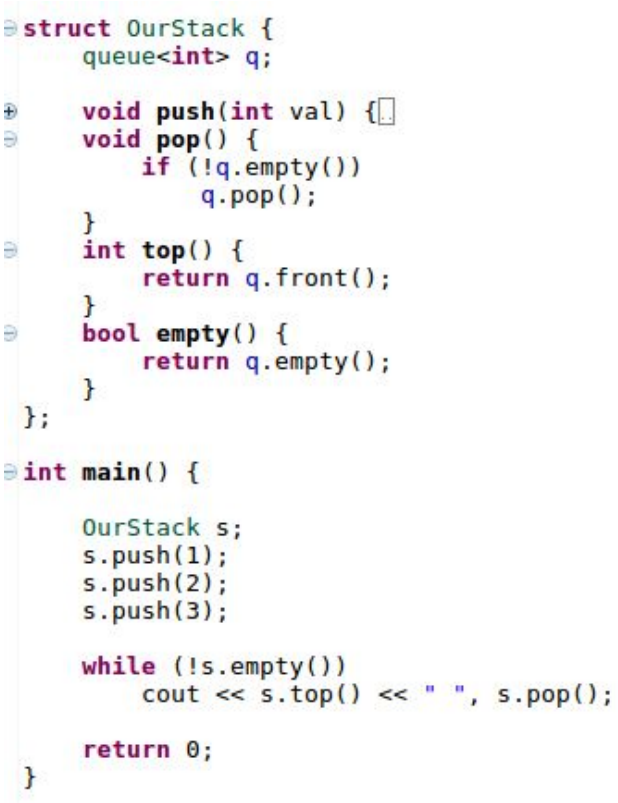
Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Homework 1: Reverse a queue using a stack

- Implement a function that reverses a queue. Use a stack for this propose.
- void reverse_queue(queue<int>& q)

Homework 2: Stack based on a queue



```
struct OurStack {
    queue<int> q;

    void push(int val) {□
    void pop() {
        if (!q.empty())
            q.pop();
    }
    int top() {
        return q.front();
    }
    bool empty() {
        return q.empty();
    }
};

int main() {

    OurStack s;
    s.push(1);
    s.push(2);
    s.push(3);

    while (!s.empty())
        cout << s.top() << " ", s.pop();

    return 0;
}
```

- We need to implement stack
- But we will make use of the available queue data structure
- Fill the push methods such that the struct behave as it is a stack
- Don’t define other struct variables. Only this queue<int>
- E.g. this code should print: 3 2 1

Homework 3: Reverse a number using stack

- Implement a method that takes number >= 0 and reverse its digits using a stack
- int reverse_num(int num)
- E.g. Input: 1234, Output: 4321

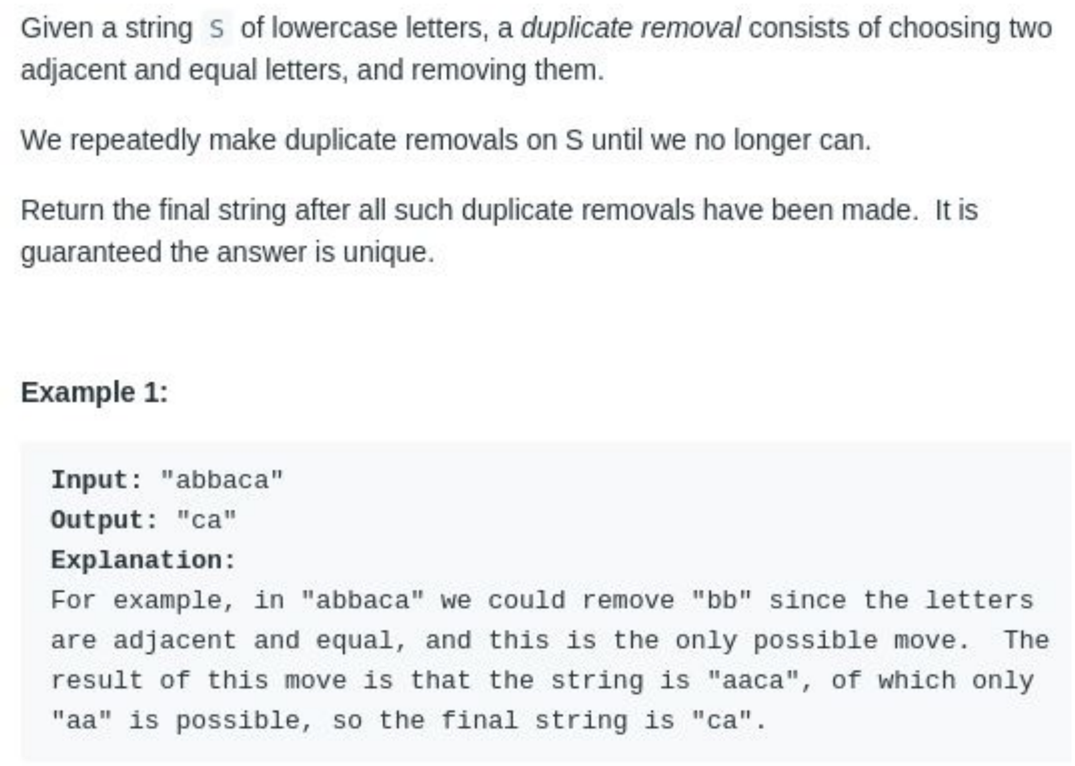
Homework 4: all words prefix (1)

- Given N, read N **unique** strings. Then Read Q queries, for each query: print all strings that start with this prefix. Print **use input order**. Use a map
- 4
  - mostafa
  - morad
  - nawal
  - marwa
- 4
  - m => [mostafa morad marwa]
  - mo => [mostafa morad]
  - mos => [mostafa]
  - naw => [nawal]

Homework 5: all words prefix (2)

- Given N, read N (**!unique**) strings. Then Read Q queries, for each query: print all strings that start with this prefix. Print **them ordered no duplicate**.
- 5
  - mostafa
  - morad
  - nawal
  - marwa
  - mostafa
- 4
  - m => [marwa morad mostafa]
  - mo => [morad mostafa]
  - mos => [mostafa]
  - naw => [nawal]

Homework 6: Remove All Adjacent Duplicates In String



Given a string `S` of lowercase letters, a *duplicate removal* consists of choosing two adjacent and equal letters, and removing them.

We repeatedly make duplicate removals on S until we no longer can.

Return the final string after all such duplicate removals have been made. It is guaranteed the answer is unique.

Example 1:

```
Input: "abbaca"
Output: "ca"
Explanation:
For example, in "abbaca" we could remove "bb" since the letters
are adjacent and equal, and this is the only possible move.  The
result of this move is that the string is "aaca", of which only
"aa" is possible, so the final string is "ca".
```

- Use a stack
- string removeDuplicates(string S)

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."