Queues -2

Assignment Solutions

1. **Remove the last k elements of a queue.**

```
#include<bits/stdc++.h>
using namespace std;



int main() {
 int n, k;
 cin >> n >> k;

 queue<int> q;

 for (int i = 0 ; i < n; i++) {
  int val;
  cin >> val;
  q.push(val);
 }

 int o = n - k;
 while (o--) {
  q.push(q.front());
  q.pop();
 }

 while (k--) {
  q.pop();
 }

 while (q.size()) {
  cout << q.front() << " ";
  q.pop();
 }

}
```

2. Reverse last k elements of a queue.

```
#include<bits/stdc++.h>
using namespace std;


int main() {
 int n, k;
 cin >> n >> k;

 queue<int> q;

 for (int i = 0 ; i < n; i++) {
  int val;
  cin >> val;
  q.push(val);
 }

 queue<int> nq; // new queue

 k = n - k;

 while (k > 0 && q.size()) {
  nq.push(q.front());
  q.pop();
  k--;
 }

 swap(nq, q);

 stack<int> s;

 while (nq.size()) {
  s.push(nq.front());
  nq.pop();
 }

 while (s.size()) {
  q.push(s.top());
  s.pop();
 }

 while (q.size()) {
  cout << q.front() << " ";
  q.pop();
 }

}
```

3. Implement queue using stacks [LeetCode 232]

```
class MyQueue {
public:
    stack<int> s1, s2;

    MyQueue() {

    }

    void push(int x) {
        s1.push(x);
    }

    int pop() {
        if(s2.empty()) {
            while(!s1.empty()) {
                s2.push(s1.top());
                s1.pop();
            }
        }
        int x = s2.top();
        s2.pop();
        return x;
    }

    int peek() {
        if(s2.empty()) {
            while(!s1.empty()) {
                s2.push(s1.top());
                s1.pop();
            }
        }
        int x = s2.top();
        return x;
    }

    bool empty() {
        if(s1.empty() and s2.empty()) return true;
        else return false;
    }
};
```

PW SKILLS

THANK YOU !