

자료구조

201818716

숙제3 과제

컴퓨터공학부

원형스택

김용현

〈토이 편집기 Undo Redo〉

```
1  /* 토이 편집기 Undo Redo */
2  #include <iostream>
3  #include <deque>
4  #include <stack>
5  #include <string>
6  using namespace std;
7
8  int main(int argc, char* argv[]) {
9      /* Faster */
10     ios::sync_with_stdio(0);
11     cin.tie(0); cout.tie(0);
12
13     /* Init */
14     deque<char> u; // Undo Deque
15     stack<char> r; // Redo Stack
16     string str; // Input
17
18     /* Input */
19     getline(cin, str);
20
21     /* Calculation */
22     for (int i = 0; i < str.size(); i++) {
23         /* Ignore Blank */
24         if (str[i] == ' ')
25             continue;
26         /* U : Undo */
27         else if (str[i] == 'U') {
28             /* Undo Deque이 비어있을 경우 */
29             if (u.empty()) {
30                 cout << "ERROR" << '\n';
31                 return 0;
32             }
33             /* 연산 */
34             r.push(u.front());
35             u.pop_front();
36         }
37         /* R : Redo */
38         else if (str[i] == 'R') {
39             /* Redo Stack이 비어있을 경우 */
40             if (r.empty()) {
41                 cout << "ERROR" << '\n';
42                 return 0;
43             }
44             /* 연산 */
45             u.push_front(r.top());
46             r.pop();
47         }
48         /* a ~ z */
49         else if ('a' <= str[i] && str[i] <= 'z') {
50             /* 새로운 입력이 들어오면 Redo Stack을 비움 */
51             while (!r.empty())
52                 r.pop();
53             /* Size가 10보다 커질 경우 */
54             if (u.size() == 10)
55                 u.pop_back();
56             /* 연산 */
57             u.push_front(str[i]);
58         }
59     }
60
61     /* Output */
62     // Undo Deque
63     if (u.size() == 0)
64         cout << "EMPTY";
65     else
66         while (u.size() != 0) {
67             cout << u.front() << ' ';
68             u.pop_front();
69         }
70
71     cout << '\n';
72
73     // Redo Stack
74     if (r.empty())
75         cout << "EMPTY";
76     else
77         while (!r.empty()) {
78             cout << r.top() << ' ';
79             r.pop();
80         }
81
82     /* Return */
83     return 0;
84 }
```

〈헤더파일〉

#. STL 중 deque, stack, string을 사용하기 위하여 각각의 헤더파일을 선언하였다.

〈Faster〉

#. 싱글 쓰레드 환경에서, printf와 scanf 함수가 이용하는 stdio.h 버퍼와 cin과 cout이 이용하는 iostream의 버퍼의 동기화를 해제함으로써, 입출력 속도를 빠르게 하였다.
ios::sync_with_stdio(0)은 stdio.h와 iostream의 동기화를 해제하겠다는 뜻이다.
cin.tie(0) 및 cout.tie(0)의 경우, cin과 cout은 서로 연결되어 있어 cin을 쓰면 출력 버퍼를 비우고 입력이 발생한다. 이러한 flush(버퍼를 비우는 과정) 과정도 시간이 들기 때문에, cin과 cout의 상호 연결을 끊어주기 위하여 해당 코드를 삽입하는 것이다.

〈Init〉

#. Undo한 내용들을 담을 deque u를 선언하였다.
#. Redo한 내용들을 담을 stack r을 선언하였다.
#. Input string을 받을 변수 str을 선언하였다.

〈Input〉

#. 입력으로 주어진 값을 입력 받는다.

〈Calculation〉

〈Ignore Blank〉

#. 입력값을 받을 때, getline함수를 통해 공백을 포함하여 입력받았다. 따라서, 공백문자를 무시하기 위해 continue를 사용하였다.

〈U : Undo〉

#. Undo한 내용을 담는 deque이 비어있을 경우, pop연산을 진행할 수 없으므로 ERROR를 발생시키고 프로그램을 종료한다. (비정상적인 종료를 뜻하기 위해 return 1을 이용해도 상관없다.)
#. Deque이 비어있지 않을 경우, Redo stack에 Undo deque의 최상단 내용을 삽입하고, 해당 원소는 deque에서 삭제한다.

〈R : Redo〉

#. Redo한 내용을 담는 stack이 비어있을 경우, pop연산을 진행할 수 없으므로 ERROR를 발생시키고 프로그램을 종료한다. (비정상적인 종료를 뜻하기 위해 return 1을 이용해도 상관없다.)
#. Stack이 비어있지 않을 경우, Undo stack에 Redo stack의 최상단 내용을 삽입하고, 해당 원소는 stack에서 삭제한다.

〈a ~ z〉

#. a ~ z 사이의 문자가 들어오면, Redo stack에 존재하는 모든 내용을 삭제한다.
#. deque의 사이즈가 10보다 커지면, pop_back 연산을 수행한다.
#. a ~ z 사이의 문자는 deque에 push_front 해준다.

〈Output〉

#. Deque과 Stack에 존재하는 모든 원소들을 출력한다. 만약, Deque과 Stack의 사이즈가 0이라면 EMPTY를 출력한다.

〈Return〉

#. 0값을 반환한다.