

3. 강박증에 걸린 박팔린

시간 제한 : 2초 | 메모리 제한 : 128MB

해설

주어진 문자열에서 서로 다른 위치의 두 문자를 골라서 서로 바꿨을 때 팰린드롬 판별을 해본다.

팰린드롬이 맞다면 두 문자의 위치를 출력한다.

팰린드롬으로 바꾸는 게 불가능하다면 -1을 출력한다.

브루트 포스(Brute Force)를 통해 이중 반복문으로 모든 경우의 수를 확인할 수 있다.

주의해야 할 점은 반드시 한 번은 바뀌어야 함을 잊지 말아야 한다. 그리고 애초에 팰린드롬 문장이 입력으로 주어진다면 가장 앞에 있는 단어와 가장 뒤에 있는 단어를 바꾸면 되므로 답은 {1, (문장의 길이)}가 된다.

```
1. cin >> input; // 문자열 입력.
2. int len = input.length();
3. for(int i=0;i<len;i++){
4.     for(int j=i+1;j<len;j++){ // 2중 반복문으로 브루트 포스.
5.         string cmp = input;
6.         cmp.replace(i, 1, input.substr(j, 1));
7.         cmp.replace(j, 1, input.substr(i, 1)); // 서로 바꿈.
8.         if(is_palindrome(cmp)==1){ // 팰린드롬이라면
9.             cout << i+1 << " " << j+1; // 서로 바꾼 위치 출력. 위치는 1부터 시작.
10.            return 0; // 프로그램 종료.
11.        }
12.    }
13. }
14. cout << -1;
15. // 위 반복문을 탈출했다면 두 문자를 바꿨을 때 팰린드롬임을 만족하는 경우가 없으므로 불가능임을 -
    1로 나타낸다.
```

시간 복잡도 $O(N^2)$ 로 해결 가능하다.