21/01/14

1. 조이스틱

**문제 설명**

조이스틱으로 알파벳 이름을 완성하세요. 맨 처음엔 A로만 이루어져 있습니다.  
ex) 완성해야 하는 이름이 세 글자면 AAA, 네 글자면 AAAA

조이스틱을 각 방향으로 움직이면 아래와 같습니다.

▲ - 다음 알파벳

▼ - 이전 알파벳 (A에서 아래쪽으로 이동하면 Z로)

◀ - 커서를 왼쪽으로 이동 (첫 번째 위치에서 왼쪽으로 이동하면 마지막 문자에 커서)

▶ - 커서를 오른쪽으로 이동

예를 들어 아래의 방법으로 JAZ를 만들 수 있습니다.

- 첫 번째 위치에서 조이스틱을 위로 9번 조작하여 J를 완성합니다.

- 조이스틱을 왼쪽으로 1번 조작하여 커서를 마지막 문자 위치로 이동시킵니다.

- 마지막 위치에서 조이스틱을 아래로 1번 조작하여 Z를 완성합니다.

따라서 11번 이동시켜 "JAZ"를 만들 수 있고, 이때가 최소 이동입니다.

만들고자 하는 이름 name이 매개변수로 주어질 때, 이름에 대해 조이스틱 조작 횟수의 최솟값을 return 하도록 solution 함수를 만드세요.

제한 사항

* name은 알파벳 대문자로만 이루어져 있습니다.
* name의 길이는 1 이상 20 이하입니다.

입출력 예

| **name** | **return** |
| --- | --- |
| JEROEN | 56 |
| JAN | 23 |

* 내 답

#include <string>

#include <vector>

#include <iostream>

using namespace std;

int solution(string name) {

int answer = name[0] > 'N' ? 'Z'-name[0]+1 : name[0]-'A';

int a\_index\_f=0;

int a\_index\_b=name.size();

int a\_num=0;

int a\_index\_f\_t = 0;

int a\_index\_b\_t = name.size();

int a\_num\_t=0;

int max\_a\_num = 0;

if(name.size()==1)

return answer;

for(int i=0;i<name.size();i++)

if(name[i]=='A')

a\_num++;

if(a\_num==name.size()) {

cout<<0<<endl;

return 0;

}

for(int i=2;i<name.size()-1;i++) {

if(name[i]=='A') {

if(name[i+1]!='A')

if(a\_num\_t+1 > max\_a\_num) {

max\_a\_num = a\_num\_t+1;

a\_index\_f\_t = i-a\_num\_t;

a\_index\_b\_t = i;

a\_num\_t = 0;

continue;

}

a\_num\_t++;

}

}

int compare\_num = a\_index\_f\_t-1 > name.size()-1-a\_index\_b\_t ? (name.size()-1-a\_index\_b\_t)\*2+a\_index\_f\_t-1 : (a\_index\_f\_t-1)\*2+name.size()-1-a\_index\_b\_t;

cout<<compare\_num<<endl;

cout<<name.size()-1<<endl;

if(!((name[1]=='A' && a\_index\_f\_t==2) || (name[name.size()-1]=='A' && a\_index\_b\_t==name.size()-2)))

if(compare\_num<name.size()-1 && max\_a\_num!=0) {

cout<<"here"<<endl;

for(int i=1;i<a\_index\_f\_t;i++)

answer+= (name[i] > 'N' ? 'Z'-name[i]+1 : name[i]-'A');

for(int i=a\_index\_b\_t+1;i<name.size();i++)

answer+= (name[i] > 'N' ? 'Z'-name[i]+1 : name[i]-'A');

answer += compare\_num;

cout<<answer<<endl;

return answer;

}

for(int i=1;i<name.size();i++) {

if(name[i]!='A')

break;

a\_index\_f++;

}

for(int i=name.size()-1;i>0;i--) {

if(name[i]!='A')

break;

a\_index\_b--;

}

if(a\_index\_f <= name.size()-a\_index\_b)

for(int i=1;i<a\_index\_b;i++)

(++answer) += (name[i] > 'N' ? 'Z'-name[i]+1 : name[i]-'A') ;

else

for(int i=name.size()-1;i>a\_index\_f;i--)

(++answer) += (name[i] > 'N' ? 'Z'-name[i]+1 : name[i]-'A');

cout<<answer<<endl;

return answer;

}

훨씬 간단하게 표현할 수 있는 방법이 있는 것 같다. 세세한 부분까지 놓치지 않기를 주의해야겠다. 조건 연산문 보다 min 함수를 사용하면 더 간단하게 표현할 수 있다.

* 남의 답

#include <string>

#include <vector>

#include <algorithm>

using namespace std;

int LUT[] = { 0,1,2,3,4,5,6,7,8,9,10,11,12,13,12,11,10,9,8,7,6,5,4,3,2,1 };

int solution(string name) {

int answer = 0;

for (auto ch : name)

answer += LUT[ch - 'A'];

int len = name.length();

int left\_right = len - 1;

for (int i = 0; i < len; ++i) {

int next\_i = i + 1;

while (next\_i < len && name[next\_i] == 'A')

next\_i++;

left\_right = min(left\_right, i + len - next\_i + min(i, len - next\_i));

}

answer += left\_right;

return answer;

}

정말 너무 간단하다. 어떻게 이렇게 생각해낼 수 있나 싶다. LUT 배열 설정하는 것도 더 간단하게 만드는 과정인 것 같다.