



INC REPORT PROBLEM A LC07

AOL Algorithm Design Analysis

Kelompok: C Level

Hanif Abdurrahman Ayash - 2702314611

Pilar Nalendra Sarwanto - 2702362604

Zenia Nadia Rifaniputri - 2702343832

UNIVERSITAS BINA NUSANTARA
JAKARTA
2024

Problem A

Problem C

The English language is infamous for having an inconsistent spelling system. One feature in the spelling system that you are interested in is the usage of the letter *c*. The letter *c* can be pronounced in multiple different ways, which might be confusing for the speaker. However, there are some simplified patterns to help speakers determine which pronunciation to use.

- It is pronounced as ⟨k⟩ if it is either: followed by *a*, *o*, or *u*; followed by a consonant letter other than *h* and *y*; or at the end of a word. The consonant letters are the English letters excluding the vowel letters: *a*, *e*, *i*, *o*, and *u*.
- It is pronounced as ⟨s⟩ if it is followed by *e*, *i*, or *y*.
- The pair of adjacent letters *ch* is pronounced with a unique sound ⟨c⟩.

You are given N words: S_1, S_2, \dots, S_N . For each given word, simplify the spelling system by replacing every occurrence of *c* into *k* or *s* (for the sound ⟨k⟩ and ⟨s⟩ respectively) according to the patterns above and every occurrence of *ch* into *c*.

Input

The first line consists of an integer N ($1 \leq N \leq 100$).

Each of the next N lines consists of a string S_i . String S_i consists of only lowercase English letters, and its length is between 1 and 100.

Output

For each word, output a single line containing a string that represents the simplified spelling system of string S_i according to the patterns above.

Sample Input #1

```
10
cyan
circle
ceramic
coccyx
delicacy
chacha
coach
blackjack
church
hiccup
```

Sample Output #1

```
syan  
sirkle  
seramik  
koksyx  
delikasy  
caca  
koac  
blakkjakk  
curc  
hikkup
```

Sample Input #2

```
6  
caesar  
cilliam  
celtic  
character  
chef  
echelon
```

Sample Output #2

```
kaesar  
silliam  
seltik  
carakter  
cef  
ecelon
```

Sample Input #3

```
4  
cacbccddcefcgchcicjckclcmcnccocpcqrcsctcucvcwxcycz c  
abdefghijklmnopqrstuvwxyz  
cch  
cc
```

Sample Output #3

```
kakbkkkdsekfgcsikjkkklmknkokpkqkrksktkukvkwxsykzk  
abdefghijklmnopqrstuvwxyz  
kch  
kk
```

Objektif Soal:

Program ini bertujuan untuk memproses beberapa kata dalam suatu input, mengubah huruf tertentu berdasarkan aturan ejaan, lalu mencetak hasilnya. Secara spesifik, huruf 'c' diproses menjadi huruf lain berdasarkan huruf yang mengikutinya, sementara huruf lainnya tetap sama.

```
#include <stdio.h>
#include <string.h>

void spelling(char *kata, char *result, int length) {
    int flag = 0;

    for (int i = 0; i < length; i++) {
        if (kata[i] == 'c') {
            if ((kata[i + 1] == 'e' || kata[i + 1] == 'i' || kata[i + 1] == 'y') && i + 1 < length) {
                result[flag++] = 's';
            }
            else if (kata[i + 1] == 'h' && i + 1 < length) {
                result[flag++] = 'c';
                i++;
            }
        }
    }
}
```

Dimulai dengan mencantumkan library yang dibutuhkan, karena pada case soal ini hanya bertujuan untuk memodifikasi string maka kami hanya menggunakan <string.h>. Untuk input output seperti biasa kami menggunakan <stdio.h>

Pada Function spelling di sinilah tempat modifikasi string terjadi terhadap beberapa huruf spesifik, dengan parameter yang dipassing ada 3 yaitu char *kata, char *result, dan int length. Langkah pertama dalam Function spelling di sini kami mendeklarasikan int flag = 0 sebagai penanda posisi karakter pada array result

```
for (int i = 0; i < length; i++) {
    if (kata[i] == 'c') {
        if ((kata[i + 1] == 'e' || kata[i + 1] == 'i' || kata[i + 1] == 'y') && i + 1 < length) {
            result[flag++] = 's';
        }
        else if (kata[i + 1] == 'h' && i + 1 < length) {
            result[flag++] = 'c';
            i++;
        }
        else if (i + 1 >= length || kata[i + 1] == 'a' || kata[i + 1] == 'o' || kata[i + 1] == 'u' || (kata[i + 1] != 'a' && kata[i + 1] != 'e' && kata[i + 1] != 'i' && kata[i + 1] != 'o' && kata[i + 1] != 'u' && kata[i + 1] != 'h' && kata[i + 1] != 'y')) {
            result[flag++] = 'k';
        }
    }
    else {
        result[flag++] = kata[i];
    }
}
result[flag] = '\0';
}
```

For loop dengan berbagai case if, else if, dan else yang saling melakukan kalkulasi untuk mengubah perubahan huruf dan pengecekan berdasarkan kasus yang diberikan. Setelah loop selesai tandai dengan \0 sebagai terminal akhir

Untuk penjabarannya kasus spesial yang dibicarakan sedari tadi ialah ketika:

- Jika karakter berikutnya adalah 'e', 'i', atau 'y' (dan berada dalam batas panjang string), 'c' diubah menjadi 's'.
- Jika karakter berikutnya adalah 'h' (dalam batas panjang string), 'c' tetap menjadi 'c' dan karakter 'h' dilewati.

- Jika tidak ada karakter setelah 'c' (atau karakter berikutnya bukan 'e', 'i', 'y', 'h'), 'c' diubah menjadi 'k'.
- Jika bukan c karakter langsung disalin ke array result

```
int main() {
    char sentence[101][105];
    int n;
    scanf("%d", &n);

    for (int i = 0; i < n; i++) {
        scanf("%s", sentence[i]);
    }

    for (int i = 0; i < n; i++) {
        char result[105];
        int length = strlen(sentence[i]);
        spelling(sentence[i], result, length);
        printf("%s\n", result);
    }

    return 0;
}
```

Pada main di sini kami membuat char array bernama sentence serta input int variabel n untuk jumlah berapa banyak sentence yang akan diinput (prosesnya terjadi dalam looping pertama). Selanjutnya panjang kata dihitung dengan strlen dan dimasukkan ke variabel int length, kemudian dipassing balik ke function spelling. Kemudian hasilnya dioutputkan

Time Complexity:

- Loop luar menghasilkan $O(N)$: Tidak ada yang spesial dari case ini, selain loop maka semua berjalan $O(1)$. Tapi terdapat proses sebagai berikut:
 - Loop utama di main memproses setiap kata dari array sentence sebanyak n kali (jumlah kata).
 - Fungsi spelling memproses setiap kata satu per satu, dengan panjang rata-rata kata L, sehingga butuh $O(L)$ untuk setiap kata.
 - Karena fungsi spelling dipanggil n kali (satu kali untuk setiap kata), totalnya adalah $O(n \times L)$

Contoh Output:

```
C:\Users\PILAR\Desktop\VCPC x + v
10
cyan
circle
ceramic
coccyx
delicacy
chacha
coach
blackjack
church
hiccup
syan
sirkle
seramik
koksyx
delikasy
caca
koac
blakkjakk
curc
hikkup

-----
Process exited after 16.14 seconds with return value 0
Press any key to continue . . . |
```

```
C:\Users\PILAR\Desktop\VCPC x + v
6
caesar
cillian
celtic
character
chef
echelon
kaesar
sillian
seltik
carakter
cef
ecelon

-----
Process exited after 1.457 seconds with return value 0
Press any key to continue . . . |
```

```
4
cacbccdcecfcgchcicjckclcmcnccocpcqcrscstcucvcwxcyczc
abdefghijklmnopqrstuvwxyz
cchh
cc
kakbkkkdsekfkgsikjkkklmknkokpkqkrksktkukvkwkxsykzk
abdefghijklmnopqrstuvwxyz
kch
kk

-----
Process exited after 1.27 seconds with return value 0
Press any key to continue . . . |
```