NITIN TS
241501136

Week 11

Q1) Two strings ***A*** and ***B*** comprising of lower case English letters are compatible if they are equal or can be made equal by following this step any number of times:

· Select a prefix from the string ***A*** (possibly empty), and increase the alphabetical value of all the characters in the prefix by the same valid amount. For example, if the string is ***xyz*** and we select the prefix ***xy*** then we can convert it to ***yx*** by increasing the alphabetical value by 1. But if we select the prefix ***xyz*** then we cannot increase the alphabetical value.

Your task is to determine if given strings ***A*** and ***B*** are compatible.

Input format

First line: String ***A***
Next line: String ***B***

Output format

For each test case, print ***YES*** if string ***A*** can be converted to string ***B***, otherwise print ***NO***.

Constraints

1 ≤ len(A) ≤ 1000000
1 ≤ len(B) ≤ 1000000

SAMPLE INPUT

abaca
cdbda

SAMPLE OUTPUT

YES

Explanation

The string ***abaca*** can be converted to ***bcbda*** in one move and to ***cdbda*** in the next move.

Status	Finished
Started	Monday, 23 December 2024, 5:33 PM
Completed	Monday, 25 November 2024, 12:09 PM
Duration	28 days 5 hours

Question **1**
Correct
Marked out of 1.00
Flag question

Two strings ***A*** and ***B*** comprising of lower case English letters are compatible if they are equal or can be made equal by following this step any number of times:

· Select a prefix from the string ***A*** (possibly empty), and increase the alphabetical value of all the characters in the prefix by the same valid amount. For example, if the string is ***xyz*** and we select the prefix ***xy*** then we can convert it to ***yx*** by increasing the alphabetical value by 1. But if we select the prefix ***xyz*** then we cannot increase the alphabetical value.

Your task is to determine if given strings ***A*** and ***B*** are compatible.

Input format

First line: String ***A***
Next line: String ***B***

Output format

For each test case, print ***YES*** if string ***A*** can be converted to string ***B***, otherwise print ***NO***.

Constraints

1 ≤ len(A) ≤ 1000000
1 ≤ len(B) ≤ 1000000

SAMPLE INPUT

abaca
cdbda

SAMPLE OUTPUT

YES

Explanation

The string ***abaca*** can be converted to ***bcbda*** in one move and to ***cdbda*** in the next move.

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

bool are_compatible(char a[], char b[]) {
    int lenA = strlen(a);
    int lenB = strlen(b);

    if (lenA != lenB) {
        return false;
    }

    for (int i = 0; i < lenA; i++) {
        if (a[i] > b[i]) {
            return false; // Characters in A cannot be decreased in value
        }
    }

    return true;
}

int main() {
    char a[1000001];
    char b[1000001];


    scanf("%s", a);
    scanf("%s", b);

    if (are_compatible(a, b)) {
        printf("YES\n");
    } else {
        printf("NO\n");
    }

    return 0;
}
```

	Input	Expected	Got	
✓	abaca cdbda	YES	YES	✓

Passed all tests! ✓

Q2) Danny has a possible list of passwords of Manny's facebook account. All passwords length is odd. But Danny knows that Manny is a big fan of palindromes. So, his password and reverse of his password both should be in the list.

You have to print the length of Manny's password and it's middle character.

Note: The solution will be unique.

INPUT

The first line of input contains the integer N, the number of possible passwords.
Each of the following N lines contains a single word, its length being an odd number greater than 2 and lesser than ***14***. All characters are lowercase letters of the English alphabet.

OUTPUT

The first and only line of output must contain the length of the correct password and its central letter.

CONSTRAINTS

1 ≤ N ≤ 100

SAMPLE INPUT

4
abc
def
feg
cba

SAMPLE OUTPUT

3 b

Question **2**
Correct
Marked out of 1.00
Flag question

Danny has a possible list of passwords of Manny's facebook account. All passwords length is odd. But Danny knows that Manny is a big fan of palindromes. So, his password and reverse of his password both should be in the list.

You have to print the length of Manny's password and it's middle character.

Note: The solution will be unique.

INPUT

The first line of input contains the integer N, the number of possible passwords.
Each of the following N lines contains a single word, its length being an odd number greater than 2 and lesser than ***14***. All characters are lowercase letters of the English alphabet.

OUTPUT

The first and only line of output must contain the length of the correct password and its central letter.

CONSTRAINTS

1 ≤ N ≤ 100

SAMPLE INPUT

4
abc
def
feg
cba

SAMPLE OUTPUT

3 b

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


void rev(char c[], const char a[]) {
    int len = strlen(a);

    for (int i = 0; i < len; i++) {
        c[i] = a[len-i-1];
    }

    c[len] = '\0';
}

int che(const char a[], const char b[]) {
    return strcmp(a, b);
}

int main() {
    int a;
    scanf("%d", &a);

    char** b = (char**)malloc(a*sizeof(char*));

    for (int i = 0; i < a; i++) {
        b[i] = (char*)malloc(14*sizeof(char));
        scanf("%s", b[i]);
    }

    for (int i = 0; i < a; i++) {
        char x[14];
        rev(x, b[i]);
        for (int j = 0; j < a; j++) {
            if (i == j)
                continue;
            if (che(x, b[j]) == 0) {
                printf("%ld %c", strlen(b[j]), b[j][(strlen(b[j])-1)/2]);
                return 0;
            }
        }
    }
}
```

	Input	Expected	Got	
✓	4 abc def feg cba	3 b	3 b	✓

Passed all tests! ✓

Q3) Joey loves to eat Pizza. But he is worried as the quality of pizza made by most of the restaurants is deteriorating. The last few pizzas ordered by him did not taste good :(. Joey is feeling extremely hungry and wants to eat pizza. But he is confused about the restaurant from where he should order. As always he asks Chandler for help.

Chandler suggests that Joey should give each restaurant some points, and then choose the restaurant having **maximum points**. If more than one restaurant has same points, Joey can choose the one with **lexicographically smallest** name.

Joey has assigned points to all the restaurants, but can't figure out which restaurant satisfies Chandler's criteria. Can you help him out?

Input:

First line has N, the total number of restaurants.
Next N lines contain Name of Restaurant and Points awarded by Joey, separated by a space. Restaurant name has **no spaces**, all lowercase letters and will not be more than 20 characters.

Output:

Print the name of the restaurant that Joey should choose.

Constraints:

$1 <= N <= 10^5$
$1 <= Points <= 10^6$

SAMPLE INPUT

3
Pizzeria 108
Dominos 145
Pizzapizza 49

SAMPLE OUTPUT

Dominos

Explanation

Dominos has maximum points.

Question **3**
Correct
Marked out of 1.00
Flag question

Joey loves to eat Pizza. But he is worried as the quality of pizza made by most of the restaurants is deteriorating. The last few pizzas ordered by him did not taste good :(. Joey is feeling extremely hungry and wants to eat pizza. But he is confused about the restaurant from where he should order. As always he asks Chandler for help.

Chandler suggests that Joey should give each restaurant some points, and then choose the restaurant having **maximum points**. If more than one restaurant has same points, Joey can choose the one with **lexicographically smallest** name.

Joey has assigned points to all the restaurants, but can't figure out which restaurant satisfies Chandler's criteria. Can you help him out?

Input:

First line has N, the total number of restaurants.
Next N lines contain Name of Restaurant and Points awarded by Joey, separated by a space. Restaurant name has **no spaces**, all lowercase letters and will not be more than 20 characters.

Output:

Print the name of the restaurant that Joey should choose.

Constraints:

$1 <= N <= 10^5$
$1 <= Points <= 10^6$

SAMPLE INPUT

3
Pizzeria 108
Dominos 145
Pizzapizza 49

SAMPLE OUTPUT

Dominos

Explanation

Dominos has maximum points.

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

typedef struct {
    char name[21];
    int points;
} Restaurant;

int compare(const void *a, const void *b) {
    Restaurant *restA = (Restaurant *)a;
    Restaurant *restB = (Restaurant *)b;

    if (restA->points != restB->points) {
        return restB->points - restA->points; // Descending order of points
    }
    return strcmp(restA->name, restB->name); // Lexicographical order of names
}

int main() {
    int n;
    scanf("%d", &n);

    Restaurant *restaurants = (Restaurant *)malloc(n * sizeof(Restaurant));
    if (restaurants == NULL) {
        printf("Memory allocation failed\n");
        return 1;
    }

    for (int i = 0; i < n; i++) {
        scanf("%s %d", restaurants[i].name, &restaurants[i].points);
    }

    qsort(restaurants, n, sizeof(Restaurant), compare);

    printf("%s\n", restaurants[0].name);

    free(restaurants);
    return 0;
}
```

	Input	Expected	Got	
✓	3 Pizzeria 108 Dominos 145 Pizzapizza 49	Dominos	Dominos	✓

Passed all tests! ✓

Q4) These days Bechan Chacha is depressed because his crush gave him list of mobile number some of them are valid and some of them are invalid. Bechan Chacha has special power that he can pick his crush number only if he has valid set of mobile numbers. Help him to determine the valid numbers.

You are given a string "S" and you have to determine whether it is Valid mobile number or not. Mobile number is valid only if it is of length 10 , consists of numeric values and it shouldn't have prefix zeroes.

Input:

First line of input is T representing total number of test cases.
Next T line each representing "S" as described in in problem statement.

Output:

Print "YES" if it is valid mobile number else print "NO".
Note: Quotes are for clarity.

Constraints:

$1 <= T <= 10^3$
sum of string length $<= 10^5$

SAMPLE INPUT

3
1234567890
0123456789
0123456.87

SAMPLE OUTPUT

YES
NO
NO

Question **4**

Correct

Marked out of 1.00

Flag question

These days Bechan Chacha is depressed because his crush gave him list of mobile number some of them are valid and some of them are invalid. Bechan Chacha has special power that he can pick his crush number only if he has valid set of mobile numbers. Help him to determine the valid numbers.

You are given a string "S" and you have to determine whether it is Valid mobile number or not. Mobile number is valid only if it is of length 10 , consists of numeric values and it shouldn't have prefix zeroes.

Input:

First line of input is T representing total number of test cases.
Next T line each representing "S" as described in in problem statement.

Output:

Print "YES" if it is valid mobile number else print "NO".
Note: Quotes are for clarity.

Constraints:

$1 <= T <= 10^3$
sum of string length $<= 10^5$

SAMPLE INPUT

3
1234567890
0123456789
0123456.87

SAMPLE OUTPUT

YES
NO
NO

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

void check_mobile_number(char s[]) {
    if (strlen(s) != 10 || !isdigit(s[0]) || s[0] == '0') {
        printf("NO\n");
        return;
    }
    for (int i = 0; i < 10; i++) {
        if (!isdigit(s[i])) {
            printf("NO\n");
            return;
        }
    }
    printf("YES\n");
}

int main() {
    int t;
    scanf("%d", &t);
    char s[20]; // Assuming input will not exceed 20 characters including '\n' and '\0'

    for (int i = 0; i < t; i++) {
        scanf("%s", s);
        check_mobile_number(s);
    }

    return 0;
}
```

	Input	Expected	Got	
✓	3 1234567890 0123456789 0123456.87	YES NO NO	YES NO NO	✓

Passed all tests! ✓