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**

**SAMPLE OUTPUT**

YES

Explanation

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

**Answer:** (penalty regime: 0 %)

```c
#include <stdio.h>
#include<string.h>
int main()
{
    char str1[1000000],str2[1000000];
    int flag=1;
    scanf("%s",str1);
    scanf("%s",str2);
    int a=strlen(str1);
    int b=strlen(str2);
    if(a==b)
    {
        for(int i=a-1;i>=0;i--)
        {
            while(str1[i]!=str2[i])
            {
                for(int j=0;j<=i;j++)
                {
                    if(str1[j]<'z')
                    str1[j]++;
                    else
                    {
                        flag=0;
                        break;
                    }
                    if(flag==0)
                    break;
                }
            }
        }
    }
    else
    flag=0;

```

P.DINESH KUMAR
240801069
ECE-B

```
    if(flag==0)
    {
        printf("NO");
    }
    else
    printf("YES");
    return 0;
}
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | abaca<br>cdbda | YES | YES | ✓ |

Passed all tests! ✓

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

P.DINESH KUMAR
240801069
ECE-B

**SAMPLE OUTPUT**

3 b

**Answer:** (penalty regime: 0 %)

```c
#include <stdio.h>
#include <string.h>
int main()
{
    int n,flag=0;
    char temp;
    scanf("%d",&n);
    char words[n][14];
    for(int i=0;i<n;i++)
    scanf("%s",words[i]);
    char reverse[14];
    for(int i=0;i<n-1;i++)
    {
        strcpy(reverse,words[i]);
        int size=strlen(reverse);
        for(int k=0;k<size/2;k++)
        {
            temp=reverse[k];
            reverse[k]=reverse[size-k-1];
            reverse[size-k-1]=temp;
        }
        for(int j=i+1;j<n;j++)
        {
            if(strcmp(reverse,words[j])==0)
            {
                flag=1;
                break;
            }
        }
        if(flag==1)
        break;
    }
    int len=strlen(reverse);
    printf("%d %c",len,reverse[len/2]);
    return 0;
}
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | 4<br>abc<br>def<br>feg<br>cba | 3 b | 3 b | ✓ |

Passed all tests! ✓

P.DINESH KUMAR
240801069
ECE-B

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.

**Answer:** (penalty regime: 0 %)

```c
#include <stdio.h>
#include <string.h>
int main()
{
    int n;
    scanf("%d",&n);
    char res[n][21];
    int rate[n];
    for(int i=0;i<n;i++)
    {
        scanf("%s",res[i]);
        scanf("%d",&rate[i]);
    }
    int max=rate[0];
    char ans[20];
    strcpy(ans,res[0]);
    for(int i=0;i<n;i++)
    {
        if(rate[i]>max)
        {
            max=rate[i];
            strcpy(ans,res[i]);
        }
        else if(rate[i]==max)
        {
            if(strcmp(res[i],ans)<0)
            strcpy(ans,res[i]);
        }
    }
    printf("%s",ans);
    return 0;
}
```

P.DINESH KUMAR
240801069
ECE-B

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | 3<br>Pizzeria 108<br>Dominos 145<br>Pizzapizza 49 | Dominos | Dominos | ✓ |

Passed all tests! ✓

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

P.DINESH KUMAR
240801069
ECE-B

SAMPLE OUTPUT

YES

NO

NO

**Answer:** (penalty regime: 0 %)

```c
#include <stdio.h>
#include <string.h>
int main()
{
    int t;
    scanf("%d",&t);
    while(t--)
    {
        int flag=1;
        char s[1000000];
        scanf("%s",s);
        int k=strlen(s);

        if(k==10)
        {
            for(int i=0;i<10;i++)
            {
                if(s[0]=='0')
                {
                    flag=0;
                    break;
                }
                if(s[i]<'0'||s[i]>'9')
                {
                    flag=0;
                    break;
                }
            }
        }
        else
        flag=0;
        if(flag==1)
        printf("YES\n");
        else
        printf("NO\n");
    }
    return 0;
}
```

| | Input | Expected | Got | |
|---|---|---|---|---|
| ✓ | 3<br>1234567890<br>0123456789<br>0123456.87 | YES<br>NO<br>NO | YES<br>NO<br>NO | ✓ |

Passed all tests! ✓

P.DINESH KUMAR
240801069
ECE-B