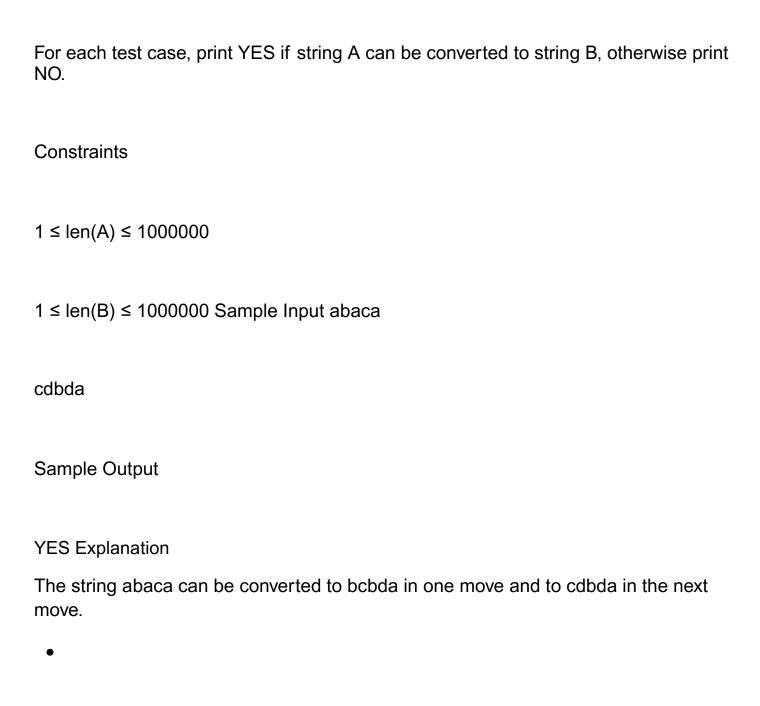
MOWNISWARAN.R ECE-D 240801209 **Problem Statement:1** Two strings A and B comprising of lower-case English letters are compatible if they are equal or can be made equal by following this step any number of times: • Select a prefix from the string A (possibly empty), and increase the alphabetical value 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

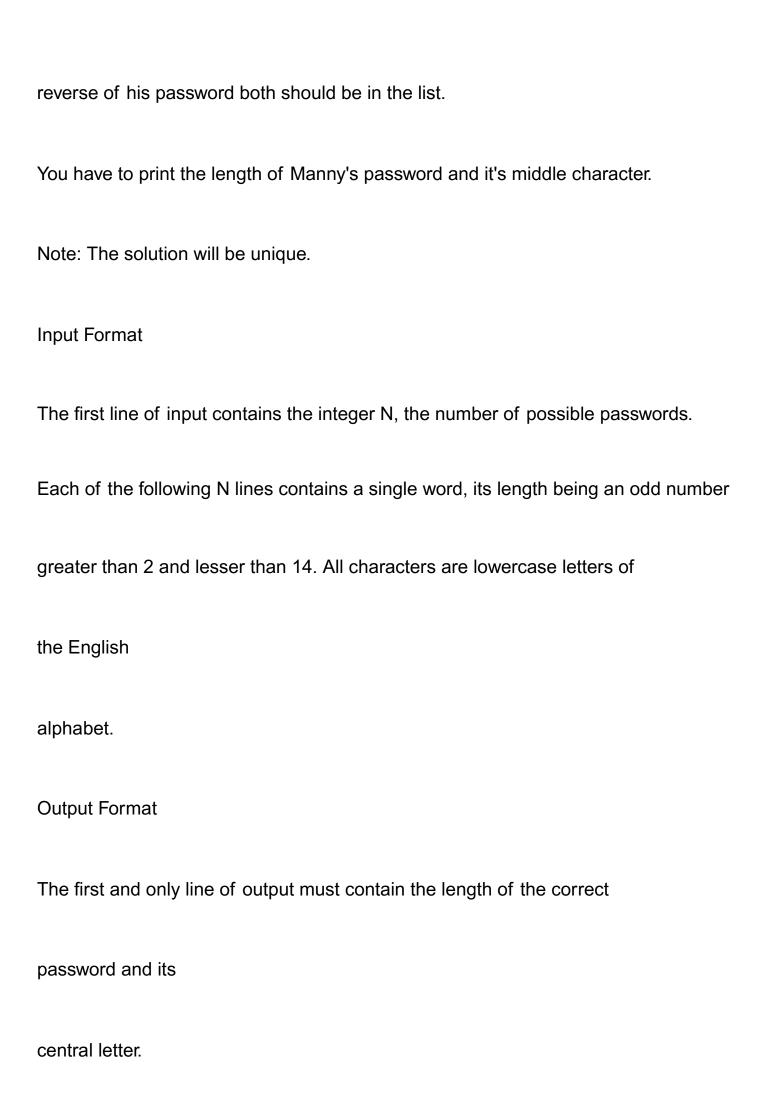


```
#include<stdio.h>
      #include<string.h>
   3
      int main()
   4 + {
          char str1[1000000],str2[1000000];
  5
          int flag=1;
   6
          scanf("%s",str1);
scanf("%s",str2);
   7
   8
          int a=strlen(str1);
          int b=strlen(str2);
  10
  11
          if(a--b)
  12 +
  13
               for(int i=a-1;i>=0;i--)
  14 +
  15
                   while(str1[i]!=str2[i])
  16 +
  17
                       for(int j=0;j<=i;j++)
  18 +
  19
                           if(str1[j]<'z')
  20
                           str1[j]++;
  21
  22 4
  23
                                flag=0;
  24
                                break;
  25
                            if(flag==0)
  26
  27
                           break;
  28
  29
  30
  31
  32
          else
  33
           flag=0;
  34
           if(flag==0)
           printf("NO");
  35
  36
           else
          printf("YES");
  37
          printf("YES");
 37
 38
          return 0;
 39 }
      Input Expected Got
      abaca
                        YES
      cdbda
Passed all tests! <
```

Problem Statement:2

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

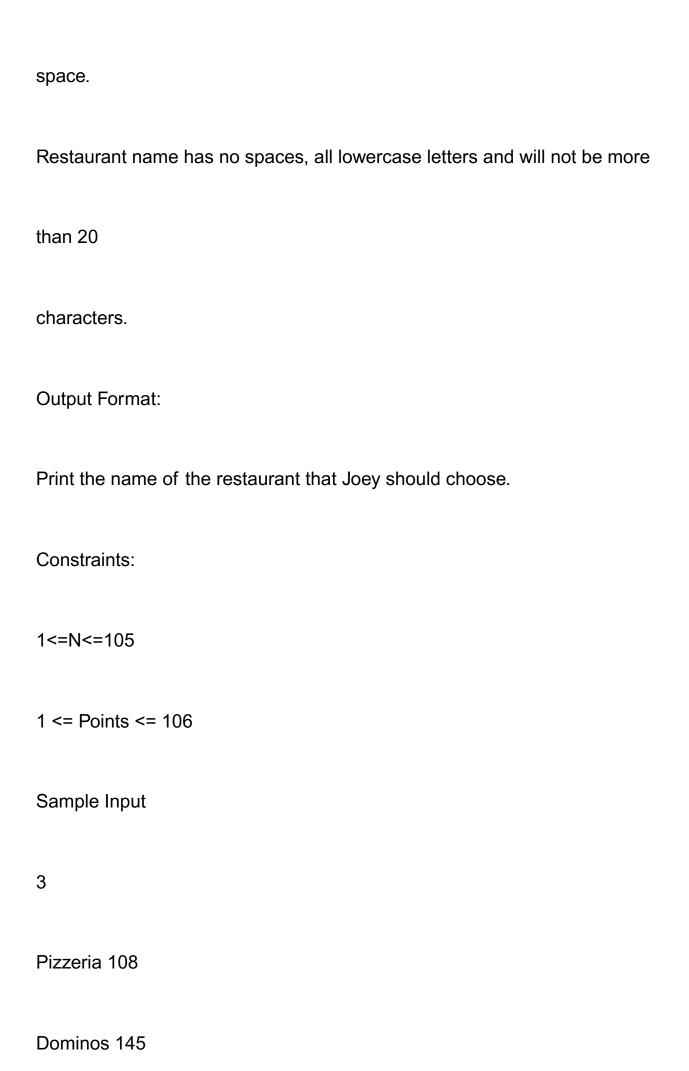


Constraints
1≤N≤100
Sample Input
4
abc
def
feg
cba
Sample Output
3 b
•

```
1
    #include<stdio.h>
 2
    #include<string.h>
 3
    int main()
 4
    {
 5
         int n,flag=0;
         char temp;
 6
         scanf("%d",&n);
 7
 8
         char words[n][14];
 9
         for(int i=0;i<n;i++)</pre>
10
         scanf("%s",words[i]);
11
         char reverse[14];
         for(int i=0;i<n-1;i++)</pre>
12
13 v
14
             strcpy(reverse,words[i]);
             int size=strlen(reverse);
15
             for(int k=0;k<size/2;k++)
16
17
18
                 temp=reverse[k];
19
                 reverse[k]=reverse[size-k-1];
20
                 reverse[size-k-1]=temp;
21
22
             for(int j=i+1;j<n;j++)</pre>
23 1
24
                  if(strcmp(reverse,words[j])==0)
25
26
                      flag=1;
27
                      break;
28
29
```

```
30
    if(flag==1)
    break;
32    }
33    int len=strlen(reverse);
34    printf("%d %c",len,reverse[len/2]);
35    return 0;
36    }
```

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 wherehe 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 Format:
First line has N, the total number of restaurants.
Next N lines contain Name of Restaurant and Points awarded by Joey,
separated by a



Pizzapizza 49

Sample Output

Passed all tests! <

Dominos

```
#include<stdio.h>
     #include<string.h>
 2
 3
     int main()
 4 ,
 5
         int n;
         scanf("%d",&n);
 6
         char res[n][21];
 7
  8
         int rate[n];
         for(int i=0;i<n;i++)
  9
 10 4
         {
              scanf("%s",res[i]);
11
12
              scanf("%d",&rate[i]);
13
         }
14
         int max=rate[0];
         char ans[20];
15
16
         strcpy(ans,res[0]);
 17
          for(int i=1;i<n;i++)
18
19
              if(rate[i]>max)
20 1
21
                  max=rate[i];
 22
                  strcpy(ans,res[i]);
23
           else if(rate[i]==max)
24
25 +
               if(strcmp(res[i],ans)<0)</pre>
26
               strcpy(ans,res[i]);
27
28
29
30
       printf("%s",ans);
31 }
   Input
                 Expected Got
                 Dominos
                          Dominos 🗸
   Pizzeria 108
   Dominos 145
   Pizzapizza 49
```



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

1<= T <= 103
sum of string length <= 105
Sample Input
3
1234567890

0123456789

0123456.87

Sample Output

YES

NO

NO

```
#include(stdio.h>
#include(string.h>
int main()
 4 + {
          int t;
scanf("%d",&t);
while(t--)
 6
 8 +
           {
               int flag=1;
char s[100000];
scanf("%s",s);
int k=strlen(s);
 9
10
11
12
13
                if(k==10)
14
                {
15
                      for(int i=0;i<10;i++)
16
17
                           if(s[0]=='0')
18
19
                                flag=0;
20
                                break;
21
                           if(s[i]<'0'||s[i]>'9')
22
23
24
25
                                flag=0;
                                break;
26
27
28
                }
29
```

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

Passed all tests! ✓