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:

$1 <= T <= 10^2$

sum of string length <= 10^5

SAMPLE INPUT

3

1234567890

0123456789

0123456.87

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[100000];

        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	
✓	abaca cdbdc	yes	yes	✓

Passed all tests! ✓

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

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\n", len, reverse[len / 2]);

    return 0;
}
```

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 = 1; 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;
}
```

	Input	Expected	Got	
✓	3 Pizzeria 188 Dominos 145 Pizzapizza 49	Dominos	Dominos	✓

Passed all tests! ✓

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:

$1 <= T <= 10^5$

sum of string length $<= 10^5$

SAMPLE INPUT

3

1234567890

0123456789

0123456.87

SAMPLE OUTPUT

YES

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

$1 \leq N \leq 100$

SAMPLE INPUT

4

abc

def

feg

cba

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\n", len, reverse[len / 2]);

    return 0;
}
```

```c
    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 = 1; 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;
}
```

	Input	Expected	Got	
✓	3 Pizzeria 100 Dominos 145 Pizzapizza 49	Dominos	Dominos	✓

Passed all tests! ✓

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:

$1 <= T <= 10^2$

sum of string length $<= 10^5$

SAMPLE INPUT

3

1234567890

0123456789

0123456.87

SAMPLE OUTPUT

YES

NO

NO