Dicotomica tracciata

Esaminate il programma che trovate in calce, rilevate gli errori e correggeteli.

Il programma deve leggere da standard input:

- un intero *n*: (potete assumere che sia compreso fra 1 e 100),
- una sequenza di esattamente *n* stringhe in ordine alfabetico crescente, scritte ciascuna su una riga,
- `y`, una delle stringhe della sequenza.

Il programma deve memorizzare la sequenza delle stringhe in un array ed effettuare una ricerca dicotomica di `y` nell'array. Ogni volta che confronta `y` con una delle stringhe, diciamo `x`, il programma deve stampare

- `> x` se `y` viene dopo `x` nell'ordine alfabetico,
- `< x` se `y` viene prima di `x` nell'ordine alfabetico,
- `= x` se `y` coincide con `x`, cosa che avverra' una volta sola alla fine.

NB: Non potete fare assunzioni sulla lunghezza delle stringhe.

Esempio 1

Eseguendo

```
    ./soluzione
```

e avendo nel flusso di ingresso

```
10
evfhyoa
evreetkt
fxvju
hakbvhb
mlf
mpbzk
pav
qxsdyyp
```

```
vymkid
wimiqvymd
hakbvhb
```

il programma emette sul flusso di uscita

```
< mlf
> evreetkt
> fxvju
= hakbvhb
```

Infatti la stringa y = "hakbvhb" viene confrontata

- innanzitutto con la stringa "mlf" (circa a meta' della sequenza), la quale viene dopo y nell'ordinamento alfabetico ---> la ricerca si restrigera' alle stringhe comprese tra la prima, "evfhyoa", e quella che precede "mlf", ovvero "hakbvhb";
- poi con la stringa "evreetkt" (circa a meta' fra "evfhyoa" e "hakbvhb"), la quale viene prima di y, ---> la ricerca si restringera' alle stringhe comprese tra quella successiva a "evreetkt", ovvero "fxvju", e "hakbvhb";
- poi con la stringa "fxvju" (circa a meta' tra "fxvju", e "hakbvhb"), la quale viene prima y, ---> la ricerca si restringera' alla sola stringa "hakbvhb";
- infine con la stringa "hakbvhb", terminando la ricerca.

Esempio 2

Eseguendo

```
    ./soluzione
```

e avendo nel flusso di ingresso

```
10
fxvju
evfhyodfa aspo  aodijad cpoja asdowepfb aodij coijw 31rsdoc  qwepdoc adpa
evreeasd awdè09knad as0nwd acpo wdc awdtkt
hakasdalskd adoianc nmasdpo asd bvhb
mlf
mpbzk
pav
qxsdyyp
```

```
vymkid
wimiqvymd
pav
```

il programma emette sul flusso di uscita

```
> mlf
< qxsdyyp
> mpbzk
= pav
```

Programma da correggere

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



char *readline() {
    char *s, *tmp;
    char c;
    int size = 2;
    int i = 0;
    s = malloc ( size );

    while ( ( c = getchar() ) != '\n') {
        if ( i > size) {
            size *= 2;
            tmp = realloc ( s, size );
            if ( tmp == NULL )
                exit(EXIT_FAILURE);
            s = tmp;
        }
        s[i] = c++;

    }
    printf( "%s\n", s );
    return s;
}



int main() {
    int n, i, left, right, mid;
    char *x[100], *y;
```

```
    scanf("%d ",&n);

    for (i=0;i<n;i++)
        x[i] = readline();


    y = readline();

    left=0; right=n-1;
    while (left<=right) {
        mid=(left+right)/2;
        if ( strcmp( x[mid], y) == 0 ) {
            printf("= %s\n",x[mid]);
            return 0;
        } else if (strcmp(x[mid],y)< 0 ) {
            printf("> %s\n",x[mid]);
            left=mid+1;
        } else {
            printf("< %s\n",x[mid]);
            right=mid-1;
        }
    }
}
```