

# Anniversaire du collègue

C'est l'anniversaire d'un collègue chez **Weekendesk** et comme ce collègue est fan de babyfoot, vous souhaitez organiser un petit tournoi avec un cadeau à la clé.

Etant donné que c'est son anniversaire, vous voulez essayer de vous arranger pour qu'il gagne le tournoi. Pour cela, vous avez donné une note à tous les participants en fonction de leur niveau (plus le score est élevé, meilleur est son niveau), et vous avez également noté votre collègue.

Le tournoi contient N joueurs, N étant une puissance de 2 afin que le tableau du tournoi soit équilibré. Par exemple, si il y a N = 8 joueurs, il y a 3 tours : les 1/4, les 1/2 et la finale. A chaque rencontre, le 1er joueur de la liste joue contre le 2ème, le 3ème contre le 4ème etc... Vous prenez l'hypothèse que c'est le joueur le plus élevé qui gagne la rencontre.

Vous avez dessiné le tableau initial des rencontres en plaçant tous les joueurs sauf votre collègue, et il vous reste à le placer. Vous cherchez à le placer de sorte qu'il rencontre le moins possible d'adversaires plus forts que lui.

Par exemple, si la liste initiale est la suivante (N= 8 donc,  $8 - 1 = 7$  joueurs car votre collègue n'est pas encore placé): 9;12;8;14;10;14;16

et que votre collègue a une note de 13, il rencontrera au mieux 1 adversaire plus fort que lui. Par exemple en se plaçant entre le 8 et le 14 :

Tour1	Tour2	Tour3
9		
12	12	
8		
13	13	13
14	14	16
10		
14		
16	16	
OK	OK	Plus fort !

En fonction de la liste des participants fournie en entrée avec leurs scores respectifs, vous devez écrire un programme permettant de déterminer le nombre minimum d'adversaires plus forts que lui qu'il devra rencontrer.

### Entrée de la fonction

La **1ère ligne** contient un entier positif N représentant le nombre total de joueurs du tournoi, tel que  $N \leq 65536$ . N est une puissance de 2 afin d'avoir un tableau équilibré.

Ex : 8

La **2ème ligne** contient le score que vous donnez à votre collègue sous la forme d'un entier compris entre 0 et 100 inclus

Ex : 13

La **3ème ligne** contient le tableau initial avant que votre collègue soit placé, soit N-1 nombres entiers compris entre 0 et 100 inclus, séparés par des points-virgules.

Ex : 9;12;8;14;10;14;16

### Sortie de la fonction

Votre fonction doit retourner un nombre entier positif représentant le nombre minimum d'adversaires plus fort que lui votre collègue devra rencontrer.

## Exemple

### Entrée

```
8
13
9;12;8;14;10;14;16
```

### Sortie

```
1
```

### Langage : Java

**Les données d'entrée de votre fonction vous sont transmises automatiquement . Vous n'avez pas à les saisir.**

Pour lire les données en entrée : **`r.readLine();`**

Pour écrire les données en sortie : **`System.out.println();`**

Vous pouvez également utiliser votre **IDE** en local. Pensez ensuite à copier/coller votre code ci-dessous et à vérifier qu'il fonctionne.

```
import java.io.BufferedReader;
import java.io.BufferedReader;
import java.io.InputStreamReader;

class kskills{

    public static void main (String[] args) throws java.lang.Exception {

        BufferedReader r = new BufferedReader (new InputStreamReader (System.in));

        //Write your code here ...

    }

}
```

- ☒ Valeurs de l'exemple
- ☒ Tous plus fort
- ☒ Tous moins fort
- ☒ 2 participants
- ☒ Tableau complexe

*Votre code doit passer tous les cas de test avec succès.*

Valeurs de l'exemple

## Input

```
8
13
9;12;8;14;10;14;16
```

## Expected output

```
1
```

Tous plus fort

## Input

```
32
60
100;99;98;97;96;95;94;93;92;91;90;89;88;87;86;85;84;83;82;81;80;79;78;77;76;75;74;73;72;71;70
```

**Expected output**

5

Tous moins fort

**Input**

4  
99  
1; 2; 3

**Expected output**

0

2 participants

**Input**

2  
65  
66

**Expected output**

1

Tableau complexe

**Input**

64  
50  
2; 28; 30; 4; 53; 54; 27; 40; 77; 17; 51; 49; 5; 76; 85; 39; 92; 75; 93; 8; 74; 38; 83; 26; 94; 56; 81; 31; 79; 42; 12; 58; 98; 1  
5; 59; 80; 44; 97; 16; 61; 99; 32; 62; 89; 25; 64; 96; 35; 95; 67; 86; 90; 46; 72; 33; 69; 23; 91; 48; 68; 19; 34; 21

**Expected output**

4