

# Lëtzebuenger Informatiksolympiad 2016

## Qualifikationsronn

Tous les programmes doivent être réalisés sous forme d'applications console (voir remarques sur le site [www.infosolympiad.lu](http://www.infosolympiad.lu) sous la rubrique "Les questionnaires").  
Les formats des données ainsi que des résultats sont à respecter.

### TÂCHE 1

### NOTES

#### Description

Le professeur d'informatique fait systématiquement trois devoirs en classe. Pour chaque élève, la note la plus basse et la note la plus élevée sont ensuite écartées. La note finale de l'élève est la note restante.

#### Entrée et sortie du programme

##### Entrée

- La première ligne du fichier d'entrée contient le nombre d'élèves  $0 < N < 50$ .
- Les  $N$  lignes suivantes contiennent chacune exactement trois entiers différents séparés par un espace, correspondant aux notes respectives des  $N$  élèves.

##### Sortie

- Le programme affiche pour chaque élève le numéro de l'élève suivi de sa note comme indiqué ci-dessous.

#### Exemple d'exécution

Données (entrée)	Résultats (sortie)
3	Eleve 1: 42
27 50 42	Eleve 2: 48
45 48 60	Eleve 3: 30
30 20 40	

#### Pour tester votre programme

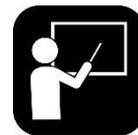
Le jeu des données de l'exemple d'exécution ci-dessus ainsi qu'un deuxième jeu de données plus complexe peuvent être téléchargés sur le site Internet [www.infosolympiad.lu](http://www.infosolympiad.lu) sous la rubrique "Les questionnaires".

**Remettez le programme sous le nom NOTES.xxx, avec xxx=PAS ou C(PP) ou JAVA.**



**Description**

Un instituteur a trouvé un moyen original pour punir les élèves dissipés. Il leur indique un nombre  $N$ . L'élève doit alors compter combien de chiffres 0, 1, 2, ..., 8, 9 figurent dans la liste des nombres 1, 2, 3, ...,  $N$ .

**Exemple**

Si  $N = 15$  alors la liste des nombres est 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

L'élève devra remettre uniquement le résultat suivant :

1 8 2 2 2 2 1 1 1 1

signifiant que le chiffre 0 apparaît 1 fois, le chiffre 1 apparaît 8 fois, le chiffre 2 apparaît 2 fois, etc.

Si le résultat est incorrect, la punition sera alourdie et l'instituteur imposera un nombre  $N$  plus grand et cela aussi longtemps jusqu'à ce que le résultat soit correct. Toto, qui est un élève bien dissipé, en a marre de ces punitions et il vous demande d'écrire un programme informatique qui calcule les résultats correctement !

**Entrée et sortie du programme****Entrée**

- La première ligne du fichier d'entrée contient  $K$ , le nombre de cas à traiter, avec  $K < 50$ .
- Les  $K$  lignes suivantes contiennent chacune exactement un entier  $N$  qu'il faut traiter avec  $0 < N < 20000$ .

**Sortie**

- Le programme affiche pour chaque  $N$  sur une ligne le nombre de chiffres 0, 1, 2, ..., 8, 9 séparés par un espace.

**Exemple d'exécution**

Données (entrée)	Résultats (sortie)
2	1 8 2 2 2 2 1 1 1 1
15	4 15 15 15 13 5 5 5 4 4
47	

**Pour tester votre programme**

Le jeu des données de l'exemple d'exécution ci-dessus ainsi qu'un deuxième jeu de données plus complexe peuvent être téléchargés sur le site Internet [www.infosolympiad.lu](http://www.infosolympiad.lu) sous la rubrique "Les questionnaires".

**Remettez le programme sous le nom PUNITION.xxx, avec xxx=PAS ou C(PP) ou JAVA.**

## Description

Dans le jargon des jeunes, les fêtes scolaires sont classées en “cool”, “moyennement cool”, “très cool”, “extrêmement cool” ou encore “exceptionnellement cool”. Comme les comités d’élèves font de plus en plus d’efforts pour rendre les fêtes de plus en plus cool, de nouveaux qualificatifs doivent régulièrement être trouvés. Cependant, tout doucement on arrive à bout de mots qualificatifs. Comme informaticien, vous savez qu’il suffit de trouver un système qui permet d’ajouter simplement des mots pour exprimer le degré de coolness. Pourquoi pas “hyper cool”, “hyper hyper cool”, “hyper hyper hyper cool”, etc.? Le système est extensible sans limites.



Écrivez un programme qui détermine combien de “hyper” doivent être utilisés. Vous aurez deux nombres en entrée: le nombre de participants estimé et le nombre de participants réel. Si le dernier est inférieur au premier, la fête a été « pas cool ». Dans le cas contraire, si le deuxième correspond à moins de 2 fois le premier la fête a été “cool”. Si le deuxième correspond au moins 2 fois le premier, mais moins de 4 fois le premier la fête a été “hyper cool”. Si le deuxième correspond au moins 4 fois le premier, mais moins de 8 fois le premier la fête a été “hyper hyper cool”. Si le deuxième correspond au moins 8 fois le premier, mais moins de 16 fois le premier la fête a été “hyper hyper hyper cool”, etc. Pour chaque doublement on ajoute une occurrence de « hyper ».

## Entrée et sortie du programme

## Entrée

- La première ligne indique le nombre  $J$  de jeux de données.
- Elle est suivie de  $J$  lignes contenant chacune deux nombres entiers :  $0 \leq E, R \leq 1\,000\,000$  sachant que  $E$  correspond au nombre de participants estimé et  $R$  au nombre de participants réel.

## Sortie

- Pour chaque jeu de données le programme affiche deux lignes. La première ligne contient “Jeu x: ”, x étant le numéro du jeu de données. Dans la deuxième ligne le programme affiche une chaîne de caractères exprimant le degré de coolness (“pas cool”, “cool”, “hyper cool”, “hyper hyper cool”, etc.).
- Après chaque jeu de données, le programme affiche une ligne vide.

## Exemple d'exécution

Données (entrée)	Résultats (sortie)
3	Jeu 1:
100 90	pas cool
1000 4001	Jeu 2:
500 890	hyper hyper cool
	Jeu 3:
	cool

## Pour tester votre programme

Le jeu des données de l'exemple d'exécution ci-dessus ainsi qu'un deuxième jeu de données plus complexe peuvent être téléchargés sur le site Internet [www.infosolympiad.lu](http://www.infosolympiad.lu) sous la rubrique "Les questionnaires".

**Remettez le programme sous le nom FETE.xxx, avec xxx=PAS ou C(PP) ou JAVA.**

## Description

En été, lorsqu'il fait très chaud, les élèves aiment bien passer leurs loisirs dans un parc d'attractions très spécial dans lequel les attractions posent le défi de surmonter des obstacles afin d'arriver à se procurer une glace.



Sur certaines attractions, il faut faire de gros efforts et surmonter plusieurs obstacles avant d'atteindre la glace, sur d'autres elle est rapidement, voire directement disponible et parfois il y a des obstacles à surmonter, mais pas de glace. Le programme à développer doit déterminer pour chaque attraction l'effort total qu'on a dû faire pour atteindre la première glace disponible sur l'attraction ou alors indiquer qu'il n'y a pas de glace sur l'attraction.

## Entrée et sortie du programme

## Entrée

- La première ligne indique le nombre  $J$  de jeux de données.
- Elle est suivie de  $J$  jeux de données. Chaque jeu de données est composé comme suit :
  - la première ligne comprend deux nombres entiers  $1 \leq A, M \leq 40$ .  $A$  désigne le nombre d'attractions et  $M$  désigne le nombre maximal d'obstacles à surmonter.
  - Elle est suivie de  $A$  lignes composée chacune de  $M$  lettres « D », « F » ou « G ». « D » signifie la présence d'un obstacle difficile coûtant un effort de 3, « F » signifie la présence d'un obstacle facile coûtant un effort de 1 et « G » signifie la présence d'une glace.

## Sortie

- Pour chaque jeu de données le programme affiche deux lignes. La première ligne contient "Jeu x :", x étant le numéro du jeu de données. Dans la deuxième ligne il affiche  $A$  positions séparées par un espace correspondant chacune à une attraction et prenant une des valeurs suivantes :
  - la somme de l'effort total nécessaire à surmonter les obstacles précédant la première glace disponible sur cette attraction,
  - la lettre « N » si aucune glace n'a été trouvée.
- Après chaque jeu de données, le programme affiche une ligne vide.

## Exemple d'exécution

Données (entrée)	Résultats (sortie)
2	Jeu 1 :
1 1	N
D	
3 5	Jeu 2 :
GGDFF	0 4 N
DFGFG	
DDDF	

## Pour tester votre programme

Le jeu des données de l'exemple d'exécution ci-dessus ainsi qu'un deuxième jeu de données plus complexe peuvent être téléchargés sur le site Internet [www.infosolympiad.lu](http://www.infosolympiad.lu) sous la rubrique "Les questionnaires".

**Remettez le programme sous le nom PARC.xxx, avec xxx=PAS ou C(PP) ou JAVA.**