

---

# COURS 1 : ALGORITHMIQUE AVANCÉE

## PROBLÈME 1 – Rendre la monnaie de sa pièce

La nouvelle version du Monopoly est sortie et on vous a attribué le passionnant rôle du banquier.

Dans la réserve de jeu, vous avez à disposition des pièces allant de 1 brouzouf à  $k$  brouzoufs, mais sans pièce à  $x$  brouzoufs (sûrement à cause d'une erreur de fabrication dans votre boîte de jeu). Toutes les autres pièces sont disponibles à volonté (la boîte de jeu est très grande).

Un joueur vous demande de la monnaie sur  $n$  brouzoufs. Pouvez-vous lui donner un assortiment de pièces dont le montant total fait  $n$  brouzoufs ?

Si plusieurs solutions sont possibles, affichez n'importe laquelle.

### Entrée

La première ligne contient un seul entier  $t$  ( $1 \leq t \leq 100$ ) — le nombre de tests.

La seule ligne pour chaque test contient 3 entiers  $n$ ,  $k$  et  $x$  ( $1 \leq x \leq k \leq n \leq 100$ ).

### Sortie

Pour chaque test, dans une première ligne, affichez "OUI" ou "NON" — selon si vous pouvez donner une quantité arbitraire de pièces de 1 à  $k$  brouzoufs, mais sans pièces à  $x$  brouzoufs, de sorte que la somme soit égale à  $n$  brouzoufs.

Si vous pouvez, la seconde ligne doit contenir un seul entier  $m$  — le nombre total de pièces que vous choisissez de donner. La troisième ligne doit contenir  $m$  entiers, séparés par des espaces, chacun étant compris entre 1 et  $k$ , mais jamais égal à  $x$ , et leur somme vaut  $n$ .

### Exemple

Entrée :

```
5
10 3 2
5 2 1
4 2 1
7 7 3
6 1 1
```

Sortie :

```
OUI
6
3 1 1 1 1 3
NON
OUI
2
2 2
OUI
1
7
NON
```

Une autre solution possible pour le premier test est  $[3, 3, 3, 1]$ . Vous ne devez pas minimiser le nombre de pièces rendues. Il existe plein d'autres solutions.

Dans le second test, vous avez juste un nombre illimité de pièces de 2 brouzoufs. Il est impossible d'obtenir 5 brouzoufs avec seulement ce montant.

Dans le dernier test, il n'y a aucune pièce de disponible du tout, donc aucun montant n'est possible.

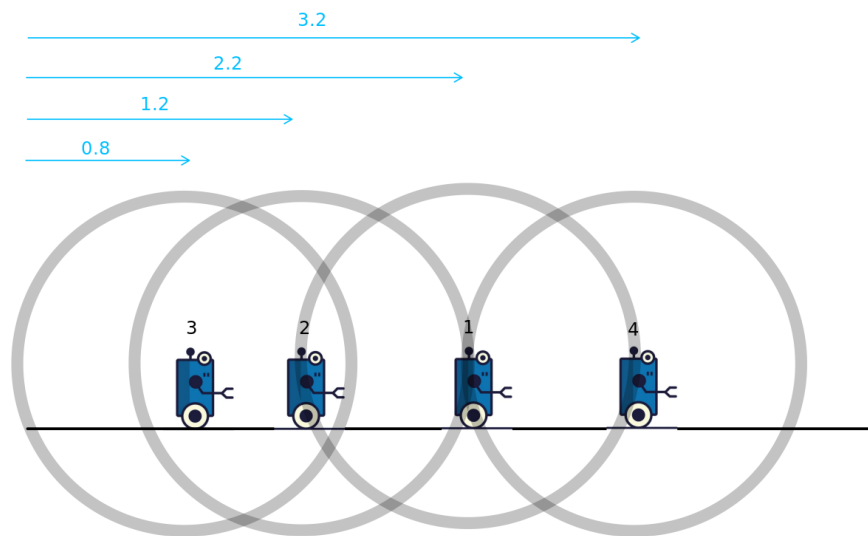
## PROBLÈME 2 – Locomotive de robots

C'est l'apocalypse : les zombies ont envahi toute la Normandie. Heureusement pour vous, vous avez à votre disposition une armée de robots qui peuvent communiquer entre eux par radio. Ces robots peuvent être pilotés à distance et se déplacent grâce à des chenilles tout-terrain.

Toutefois, la portée d'émission/réception de chaque robot est limitée à 1 kilomètre maximum. Vous pouvez transiter vos informations à un robot via un autre robot. Donc pour pouvoir télécommander un robot, il faut qu'il se situe à un kilomètre ou moins de vous, ou alors qu'il se situe à un kilomètre ou moins d'un autre robot qui peut communiquer avec vous. Autrement dit, vous pouvez télécommander une chaîne de robots du moment où deux robots qui se suivent sont séparés d'un kilomètre au maximum.

Dernière contrainte : les robots n'ont pas tous la même autonomie et peuvent se déplacer que d'un certain nombre de kilomètres. On stocke la distance maximum que peut parcourir chaque robot dans un tableau.

Avec tous ces robots, vous voulez savoir la distance maximum à laquelle vous pouvez éloigner un robot de vous. Par exemple, si vous avez quatre robots : le premier se déplaçant de 3.0 km, le deuxième de 1.2 km, le troisième de 0.8 km et le quatrième de 3.9 km, alors vous pouvez télécommander un robot (le quatrième) jusqu'à une distance de 3.2 km, comme le montre le schéma suivant :



### Entrée

La première ligne contient un seul entier  $t$  ( $1 \leq t \leq 100$ ) — le nombre de tests.

La première ligne de chaque test contient un entier  $r$  ( $1 \leq r \leq 500000$ ) — le nombre de robots.

La seconde ligne de chaque test contient  $r$  nombres flottants, correspondant aux distances maximum que les robots peuvent parcourir.

### Sortie

Pour chaque test, affichez la distance la plus lointaine qu'un de vos robots puisse atteindre, sans que vous ne bougiez de votre position actuelle.

### Exemple

Entrée :

```
2
4
3.0 1.2 0.8 3.9
5
2.3 2.1 2.2 2.5 2.0
```

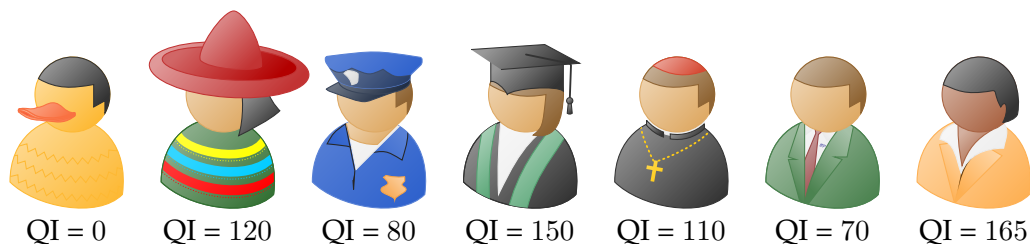
Sortie :

```
3.2
2.5
```

### PROBLÈME 3 – Il y a toujours plus idiot que soi.

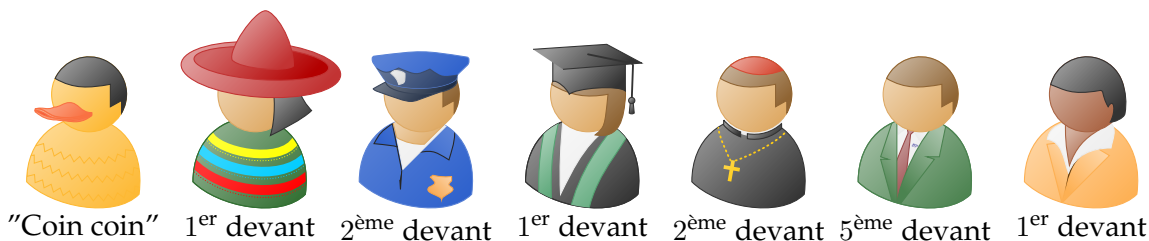
Vous conduisez une étude scientifique visant à vérifier l’adage “Il y a toujours plus idiot que soi.”. Vous sollicitez  $n$  cobayes pour votre expérience et leur demandez de se mettre à la queue leu leu.

Vous connaissez le QI de chaque personne. Pour ne vexer personne, vous avez placé un canard avec une perruque en tête de la file. Il a un QI de zéro (il n’est pas possible de faire pire). Un exemple :



À part le canard, chaque personne doit annoncer à combien de places devant elle se situe la première personne la moins intelligente qu’elle.

Dans l’exemple du dessus,



Vous devez afficher chacun des nombres ainsi criés, dans l’ordre de la file.

#### Entrée

La première ligne contient un seul entier  $t$  ( $1 \leq t \leq 100$ ) — le nombre de tests.

La première ligne de chaque test contient un entier  $n$  ( $1 \leq n \leq 100000$ ) — le nombre de cobayes (canard exclus).

La seconde ligne de chaque test contient  $n$  nombres, correspondant aux QI de chaque personne, dans l’ordre de la file (canard exclus) :  $qi[1]$   $qi[2]$   $\dots$   $qi[n]$ . On considère que  $qi[0] = 0$ .

#### Sortie

Pour chaque test et pour  $i \in \{1, \dots, n\}$ , on définit  $p[i]$  comme le nombre strictement positif tel que

- $qi[i-p[i]] < qi[i]$
- $qi[i-j] \geq qi[i]$  pour  $j \in \{1, \dots, p[i] - 1\}$ .

Pour chaque test, on affichera le tableau  $p[1]$   $p[2]$   $\dots$   $p[n]$ .

#### Exemple

Entrée :

```
3
6
120 80 150 110 70 165
5
100 120 130 180 125
8
50 140 130 120 110 100 90 80
```

Sortie :

```
1 2 1 2 5 1
1 1 1 1 3
1 1 2 3 4 5 6 7
```

---

#### PROBLÈME 4 – Double maléfique

On veut compter le nombre de fois où un élément et son double apparaissent dans le tableau.

Par exemple pour `tab = [1, 5, 20, 10, 7, 2]`, le nombre recherché est 3 car on retrouve 1 et 2, puis 5 et 10 et enfin puis 10 et 20.

Par souci de simplicité, on supposera que le tableau ne contient pas le nombre 0 et ne contient aucun élément en plusieurs exemplaires (i.e aucun doublon).

#### Entrée

La première ligne contient un seul entier  $t$  ( $1 \leq t \leq 100$ ) — le nombre de tests.

La première ligne de chaque test contient un entier  $n$  ( $1 \leq n \leq 100000$ ) — la taille du tableau.

La seconde ligne de chaque test contient les  $n$  nombres du tableau. Ils sont tous distincts deux à deux et jamais égaux à 0.

#### Sortie

Pour chaque test, affichez le nombre d'éléments  $x$  du tableau tel que  $2x$  appartient aussi au tableau.

#### Exemple

Entrée :

```
3
6
1 5 20 10 7 2
4
12 48 -24 3
5
-400 -100 -25 -50 -200
```

Sortie :

```
3
0
4
```