

## Demi-Finale

Déroulement, règlement et dates importantes de la compétition sur [www.infosolympiad.lu](http://www.infosolympiad.lu)



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 d'entrée ainsi que des résultats représentés dans les exemples d'exécution sont à respecter absolument.



Sous le fichier d'entrée on entend soit l'entrée directe des données via le clavier soit la redirection d'un fichier texte en mode console.



Les programmes n'ont pas besoin de tester d'éventuels débordements.

## TÂCHE 1

## CUISINE

20 POINTS

### Description

La cuisinière Maus Kätti fait face à une mission difficile. La tâche ardue est de faire cuire des pommes de terre pour tous les élèves de l'école des cuisiniers.

Il est bien connu, qu'avant de faire bouillir des pommes de terre, elles doivent être pelées et lavées.

À  $N$  élèves cuisiniers est attribué le devoir de peler les pommes de terre. La vitesse de pelage peut varier pour chaque élève cuisinier. Supposons que pour le  $i^{\text{ème}}$  élève cuisinier,  $A_i$  est le nombre de pommes de terre qu'il peut peler en une minute.

À  $M$  élèves cuisiniers est attribué le devoir de laver les pommes de terre.  $B_j$  indique le nombre de pommes de terre que le  $j^{\text{ème}}$  élève cuisinier peut laver en une minute. À la fin de chaque minute, Maus Kätti collecte les pommes de terre pelées et les transfère au lavage. En raison du niveau élevé de discipline, nous pouvons supposer en toute sécurité que la collecte et le transfert des pommes de terre ne prennent aucun temps.

On sait que le nombre de pommes de terre qui doivent être cuites pour le dîner est égale à  $K$ . À la fin de chaque minute, le chef de cuisine Rappid vérifie l'état d'avancement et si, au moment de sa visite, la quantité de pommes de terre pelées et lavées est supérieure ou égale à  $K$ , Rappid ordonne aux élèves cuisiniers de s'arrêter. Vous pouvez supposer que le nombre de pommes de terre disponibles est illimité.

Maus Kätti aimerait déterminer le nombre de minutes qui s'écouleront avant qu'elle ne soit relevée de ses fonctions. Votre tâche est de l'aider.



## Entrée et sortie du programme

### Entrée

La première ligne du fichier d'entrée contient trois nombres entiers  $N$  (le nombre d'élèves cuisiniers qui pèlent),  $M$  (le nombre d'élèves cuisiniers qui lavent) et  $K$  (la quantité requise de pommes de terre), séparés par un espace.

Les  $N$  lignes suivantes consistent chacune en un entier  $A_i$  ( $1 \leq A_i \leq 1\,000$ ) - le nombre de pommes de terre que le  $i^{\text{ème}}$  élève cuisinier peut peler en une minute.

Viennent ensuite  $M$  lignes, chacune composée d'un entier  $B_j$  ( $1 \leq B_j \leq 1\,000$ ) - du nombre de pommes de terre que le  $j^{\text{ème}}$  élève cuisinier peut laver en une minute

### Sortie

Le résultat doit être composé d'un seul nombre entier - le nombre de minutes précédant l'ordre d'arrêt du pelage et du lavage.

### Exemples d'exécution

Fichier d'entrée	Résultat
2 1 10 5 5 1	11
3 2 21 2 2 1 3 3	6

#### Premier exemple

À partir de la deuxième minute, une pomme de terre serait prête chaque minute. Par conséquent, 10 pommes de terre seraient pelées et lavées à la fin de la 11<sup>ème</sup> minute.

#### Deuxième exemple

À la fin de la deuxième minute, 5 pommes de terre seraient prêtes ; 3<sup>ème</sup> - 10 ; 4<sup>ème</sup> - 15 ; 5<sup>ème</sup> - 20 ; 6<sup>ème</sup> - 25 ; après cela, Rappid ordonnerait de s'arrêter.

### Contraintes

$$1 \leq N, M \leq 1\,000$$

$$1 \leq K \leq 2\,000\,000\,000$$



Remettez le programme sous le nom CUISINE.xxx, avec xxx = C(PP), PY ou JAVA.

## Description

Dans votre lycée la remise des diplômes de fin d'études est un événement important et il va de soi qu'une photo de tous les diplômés est prise. Le lycée tient à ce que sur la photo tous les diplômés soient dans la première rangée et a déjà décidé de l'ordre de placement des filles ainsi que de celui des garçons. Il souhaiterait en fait avoir un mélange aussi régulier que possible entre filles et garçons et minimiser les différences de taille (hauteur) entre élèves avoisinants. Malheureusement ces deux buts sont évidemment en conflit. Vous devrez trouver une solution algorithmique pour donner un coup de main au photographe.



Vous disposez d'une liste donnant la taille  $t_i$  de chacune des  $f$  filles ainsi que d'une liste donnant la taille  $t_j$  de chacun des  $g$  garçons. Certaines années il y a plus de filles, d'autres années plus de garçons diplômés. Parfois ces nombres seront identiques. Supposons pour le besoin d'un exemple que cette année  $f \geq g$ . Dans ce cas les garçons définissent  $g + 1$  intervalles et en moyenne il devrait y avoir  $\frac{f}{g+1}$  filles par intervalle. Bien entendu, l'expression  $\frac{f}{g+1}$  pourrait ne pas donner un entier. Si cela est le cas, vous pourrez décider pour quels intervalles l'arrondi à l'entier est fait vers le bas et pour lesquels il est fait vers le haut.

Pour minimiser les différences de taille entre élèves avoisinants, vous considérerez la formule suivante :

$$\text{différence de taille (hauteur) totale} = \sum_{k=1}^{n-1} (h_{p_{k+1}} - h_{p_k})^2$$

Il faut savoir que  $h_{p_k}$  désigne la taille (hauteur) de la personne qui se trouve à la position  $k$ .

## Entrée et sortie du programme

## Entrée

La première ligne contient deux entiers entre 1 et 500 ( $1 \leq f, g \leq 500$ ), séparés par un espace.

La deuxième ligne contient  $f$  nombres entiers  $t$  entre 0 et 1000 ( $0 \leq t \leq 1000$ ). Les nombres sont triés dans l'ordre dans lequel les filles devraient se trouver sur la photo, séparés par un espace.

La troisième ligne contient  $g$  nombres entiers  $t$  entre 0 et 1000 ( $0 \leq t \leq 1000$ ). Les nombres sont triés dans l'ordre dans lequel les garçons devraient se trouver sur la photo, séparés par un espace.

## Sortie

Le résultat représente la différence de taille totale la plus petite possible au vu des contraintes indiquées dans l'énoncé.

## Exemples d'exécution

Fichier d'entrée	Résultat
3 6 150 165 180 152 155 157 159 163 170	516
4 54 997 752 604 332 97 295 202 781 51 600 209 650 662 331 830 456 817 623 930 901 541 882 652 398 814 490 37 230 965 829 448 298 640 758 365 807 181 303 720 552 915 741 86 440 213 201 410 341 888 428 882 483 71 258 533 268 681 883	7258564



Remettez le programme sous le nom PHOTO.xxx, avec xxx = C(PP), PY ou JAVA.

**Description**

Pendant les vacances d'hiver, certaines classes du Lycée Gabriel Lippmann participent à des classes de neige en Suisse. Comme il y a beaucoup d'élèves qui se sont inscrits, le directeur du lycée doit réserver plusieurs autocars afin d'assurer le transport. Comme les élèves inscrits d'une même classe souhaitent absolument être dans le même autocar et comme la direction du lycée désire minimiser le nombre d'autocars (pour des raisons financières), écrivez un programme qui détermine le nombre minimal d'autocars nécessaires.



Chaque autocar possède une capacité assise de  $P = 50$  places. Il n'y a pas de places debout.

**Exemple**

Soient  $N = 7$  classes participantes avec les nombres d'élèves inscrits par classe suivants :

20

9

23

20

12

10

1

Le nombre minimal d'autocars nécessaires est  $M = 2$ .

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

Le fichier d'entrée contient une première ligne qui représente le nombre de classes participantes  $N$ .

Les  $N$  lignes suivantes indiquent les nombres d'élèves inscrits par classe, une classe par ligne.

**Sortie**

Le résultat représente le nombre minimal d'autocars nécessaires  $M$ .

**Exemple d'exécution**

Fichier d'entrée	Résultat
7	2
20	
9	
23	
20	
12	
10	
1	

**Contraintes**

Le nombre de classes participantes  $N$  est compris dans l'intervalle entier  $[1 ; 50]$ .

Le nombre d'élèves inscrits par classe est compris dans l'intervalle entier  $[1 ; 29]$ .

La capacité par autocar  $P$  est une constante égale à 50.



Remettez le programme sous le nom **NEIGE.xxx**, avec  $xxx = C(PP), PY$  ou **JAVA**.