

# beCP 2019

## Tâche 1.2: Planification (scheduling)

---

Auteurs: Robin Jadoul, Damien Galant    Préparation: Damien Galant  
Limite de temps: 0s    Limite mémoire: 0MB

---

*Note: Cette tâche est output only. Cela veut dire que vous devez uniquement soumettre les résultats de vos calculs. Vous ne devez pas soumettre de code. Il y a plusieurs fichiers d'input, et vous pouvez soumettre vos solutions ensemble ou séparément. Bien entendu, cela signifie que vous n'êtes pas obligés d'utiliser le même programme pour résoudre chaque input, et vous pouvez même essayer d'en résoudre quelques-uns à la main.*

Les entraîneurs beCP ont beaucoup travaillé récemment pour préparer le concours.

Il y a  $T$  entraîneurs. Il reste  $N$  tâches à compléter pour que tout soit prêt (relire les énoncés, vérifier des tests cases, uploader quelques fichiers sur le serveur, ...). C'est la veille du concours, et puisqu'il est déjà tard, ils veulent partager les actions entre eux pour minimiser l'heure à laquelle ils auront tous fini et pourront aller dormir.

Votre but est d'aider les entraîneurs en répartissant les tâches à compléter parmi les instructeurs. Il n'y a pas d'ordre spécifique à respecter entre les tâches, et n'importe quel entraîneur peut compléter n'importe quelle tâche.

### Input

La première ligne contient deux entiers  $T, N$  : le nombre d'entraîneurs et le nombre de tâches à compléter. La deuxième ligne contient  $N$  entiers strictement positifs : le temps  $d_i$  (en minutes) que chaque tâche prend à compléter.

### Output

La première ligne de l'output doit contenir un seul entier : le nombre de minutes passées entre maintenant et le moment où le dernier entraîneur finit de travailler. Vous devez ensuite écrire  $T$  lignes d'output. La  $i$ ème ligne commence avec un entier  $k \geq 0$  : le nombre de tâches assignées à l'entraîneur  $i$ . Ceci est suivi de  $k$  entiers : les durées des tâches que l'entraîneur  $i$  doit compléter.

Chaque tâche doit être assignée à exactement un entraîneur.

### Limites générales

- $1 \leq T \leq 20$ ;
- $1 \leq N \leq 20000$ ;
- $1 \leq d_i \leq 100$ ;

### Calcul des points

Pour chacun des 10 test cases, si votre programme donne un temps  $t$ , et  $t_{best}$  est le meilleur temps obtenu par le jury, alors le **ratio** de votre solution est défini par la formule suivante

$$r = \max \left( \min \left( \frac{t - t_{best}}{t_{best}}, 1 \right), 0 \right).$$

(Il est possible que vous trouviez de meilleures solutions que celles obtenues par le jury !)

Le score de votre programme est alors calculé ainsi :

$$\text{score} = 10^{1-10r}.$$

En particulier, si votre solution est au moins aussi bonne que celle trouvée par le jury, vous obtiendrez 10 points pour ce test case.

Votre score total pour ce problème est simplement la somme des scores pour chacun des test cases. Le maximum est donc 100 points.

### Exemple 1

sample1.in	sample1.out
2 5 1 2 3 4 5	8 3 3 4 1 2 5 2

Dans cet exemple, il y a 2 entraîneurs et 5 tâches. Une solution optimale possible consiste à assigner les tâches de durée 1, 3 et 4 au premier entraîneur, et les tâches de durée 2 et 5 au deuxième entraîneur. Le premier entraîneur finit de travailler après  $3 + 4 + 1 = 8$  minutes. Le deuxième entraîneur finit de travailler après  $5 + 2 = 7$  minutes. La solution est donc 8 car tous les entraîneurs ont fini de travailler après 8 minutes.

Ce n'est pas l'unique solution.

L'ordre dans lequel vous listez les tâches assignées à chaque entraîneur n'a pas d'importance.