

Le second fichier donné en paramètre devra être généré par votre programme et contenir :

- en première ligne : le nombre de tests traités
- sur chaque ligne suivante : une liste d'entiers séparés par une virgule représentant une campagne de mesure "nettoyée".

▷ **Question 2:** (5 pts) Proposez une version en temps linéaire de votre algorithme (Votre programme devra se trouver dans un fichier dénommé `P2.py`, être invocable de la même façon et avec les mêmes paramètres que ceux de la question précédente.

★ **Exercice 2:** Bonnes résolutions! (10 points)

Vous avez pris de bonnes résolutions pour 2022 en décidant de refaire de l'exercice de façon intensive. Nul besoin pour cela d'investir dans des équipements volumineux et coûteux. Les escaliers représentent par exemple, un formidable terrain d'entraînement, notamment pour le *cardio-training*.

▷ **Question 1:** (5 pts) Vous disposez d'un escalier à N marches et vous pouvez franchir à chaque étape, 1 ou 2 marches à la fois. Etant donné N et en considérant que l'ordre des actions compte, développez une fonction qui renvoie le nombre de façons différentes de monter les escaliers. Par exemple, si N est égal à 4, il y a 5 façons différentes de monter les escaliers :

- 1,1,1,1
- 2,1,1
- 1,2,1
- 1,1,2
- 2,2

Votre programme devra se trouver dans un fichier dénommé `P3.py` qui doit être invocable avec la commande suivante dans un terminal :

```
python3 P3.py E3.txt S3.txt
```

Le fichier `E3.txt` est donné. Il contient :

- en première ligne : le nombre de tests présents,
- sur chaque ligne suivante : un entier qui donne la taille en nombre de marches de l'escalier.

Le second fichier donné en paramètre devra être généré par votre programme et contenir :

- en première ligne : le nombre de tests traités
- sur chaque ligne suivante : deux entiers séparés par une virgule. Le premier indique le nombre de marches du test, le second le nombre de chemins uniques calculé par votre algorithme

▷ **Question 2:** (5 points) Vous devenez plus fort et vous pouvez maintenant grimper n'importe quel nombre de marches à la fois dans une liste L donnée. Par exemple, si $L = \{1, 2, 4, 5\}$, vous pouvez enjamber 1,2,4 ou 5 marches à la fois. Généralisez votre fonction pour intégrer cette nouvelle capacité. Votre programme devra se trouver dans un fichier dénommé `P4.py` qui doit être invocable avec la commande suivante dans un terminal :

```
python3 P4.py E4.txt S4.txt
```

Le fichier `E4.txt` est donné. Il contient :

- en première ligne : le nombre de tests présents,
- sur chaque ligne suivante : un entier qui détermine la taille en nombre de marches de l'escalier suivi d'une suite d'entiers définissant vos capacités de franchissement. Par exemple, la séquence 5,1,2,3,4 indique que l'escalier a 5 marches et que vous êtes capable de franchir 1,2,3,4 marches simultanément.

Le second fichier donné en paramètre devra être généré par votre programme et contenir :

- en première ligne : le nombre de tests traités
- sur chaque ligne suivante : deux entiers séparés par une virgule. Le premier indique le nombre de marches du test, le second le nombre de chemins uniques calculé par votre algorithme.

Bonne année 2022 !