

ATTENTION : ce travail est à faire pendant les vacances et doit être rendu sur PRONOTE pour **le lundi 8 novembre au plus tard**, même si nous n'avons cours que le mardi 9. Aucun délai ne sera accepté et entraînera la nullité de la note.

Principe

On considère le jeu suivant :

- N est un entier positif compris entre 10 et 25 (au choix ou au hasard, peu importe);
- on dispose N jetons identiques sur une table;
- deux joueurs A et B jouent à tour de rôle, A commence;
- chacun d'eux, lorsque c'est son tour, prélève soit 1, soit 2, soit 3 jetons;
- bien entendu, s'il reste 2 jetons on ne peut en prendre 3 *et cætera*;
- il est interdit de passer son tour;
- le joueur qui prélève le(s) dernier(s) jeton(s) a perdu.

Exemple

On décide de démarrer la partie avec 14 jetons.

- A commence par en prendre 1, il en reste 13;
- B en prend 3 il en reste 10;
- A en prend 1 il en reste 9;
- B en prend 2 il en reste 7;
- A en prend 2 il en reste 5;
- B en prend 3 il en reste 2;
- A en prend 1 il en reste 1;
- B en prend 1 et donc B a perdu.

Travail à faire

Premier programme

1. Au brouillon, écris toi-même au moins deux exemples de partie en choisissant N entre 10 et 25, pour bien comprendre comment le jeu se déroule.

Pour la suite de l'exercice on considère que A est l'ordinateur et B un être humain.

2. Tu vas devoir écrire en PYTHON un programme dans lequel l'ordinateur et l'humain jouent l'un contre l'autre :

- l'ordinateur joue pour l'instant au hasard, mais en veillant bien à respecter les règles;
- l'être humain fait des propositions mais le programme vérifie qu'il respecte lui aussi les règles.

Tu **devras** utiliser les variables suivantes :

- **n**, de type **int**, qui représente le nombre de jetons restants;
- **choix_ordi**, de type **int**, qui représentera à chaque tour de jeu le nombre de jetons que l'ordinateur choisit de prendre;
- **choix_humain**, de type **int**, qui représentera à chaque tour de jeu le nombre de jetons que l'humain choisit de prendre;
- **gagnant**, de type **str**, qui vaudra **"humain"** ou **"ordi"** à la fin de la partie (et que l'on affichera).

Pour t'aider, voici une description du fonctionnement du programme :





- Le programme commence par choisir un nombre de jetons au hasard;
- ensuite tant qu'il reste des jetons
 - l'ordinateur joue;
 - s'il n'y a plus de jetons le gagnant est l'humain;
 - sinon, c'est à l'humain de jouer;
 - s'il ne reste plus de jetons, c'est l'ordinateur qui gagne.
- on affiche le gagnant.

Pour choisir un nombre au hasard entre **a** et **b inclus**, il faut d'abord inclure au début de ton programme :

```
from random import randint
```

Ensuite pour choisir un nombre entre 10 et 15, tu utilisera **randint(10, 25)**.

Comment taper mon programme Python ?

1. Tu peux utiliser des éditeurs en ligne : https://www.onlinegdb.com/online_python_compiler est très bien fait, tu peux taper ton code et l'exécuter.
Une fois le code terminé, tu le copieras ( + ) et le colleras ( + ) dans un fichier texte nommé **prog1.txt**.
Si tu sais le faire, tu peux le nommer **prog1.py** mais ce n'est pas obligé.
2. Tu peux aussi installer EDUPYTHON, qui est très simple à prendre en main et qui se trouve à cette adresse : <https://edupython.tuxfamily.org/>

Tu déposeras le fichier sur PRONOTE.

Deuxième programme

Pose-toi les questions suivantes et réponds-y :

Quand il reste 4 jetons, combien faut-il en prendre pour être sûr de gagner ?
De même pour 3 jetons. De même pour 2 jetons.

Modifie ton premier programme pour que l'ordinateur joue mieux en fin de partie.

Tu l'enregistreras dans un fichier nommé **prog2.txt** ou **prog2.py** et tu déposeras sur PRONOTE.

Troisième programme

1. Donner les valeurs de n pour lesquelles A est sûr de gagner en un coup (c'est-à-dire laisser un seul jeton à B).
2. Donner les valeurs de n pour lesquelles A est sûr de gagner en deux coups, c'est-à-dire : A joue, B joue, A joue et B a obligatoirement perdu (se ramener à la question précédente).
3. En extrapolant, quel est l'ensemble \mathcal{E} des valeurs de départ de N (plus nécessairement compris entre 10 et 25) pour lesquelles A est sûr de gagner (on ne demande pas de preuve) ?
4. Quand A est sûr de gagner, quelle est la stratégie gagnante ?
5. En déduire un troisième programme qui, si $N \in \mathcal{E}$, fait en sorte que A gagne. Tu le nommeras **prog3.txt** et tu le déposeras sur pronote.