

Exercice 1-Salle de sport

Alain vient d'ouvrir un club de gym en 2017 et il a déjà 100 adhérents. D'après ses pronostics, le nombre d'adhérents devrait augmenter de 8% par an. Écrire un programme pour aider Alain dans ses prévisions de nombre d'adhérents, en suivant le cahier des charges ci-dessous.

Le nombre d'adhérents chaque année sera arrondi à l'entier le plus proche grâce à la fonction `int_round(...)`.

Lorsqu'Alain lance l'application, le menu suivant doit s'afficher :

Menu, veuillez choisir :

1. Prévisions adhérents à l'année N (résumé)
2. Prévisions adhérents à l'année N (détails)
3. Adhésions cumulées à l'année N
4. Année à laquelle on obtiendra M adhérents
- Q. Quitter

- Si Alain tape 1, le programme lui affiche **Choisissez une année** : puis le nombre d'adhérents qu'il y aura à l'année choisie.
Exemple : si Alain tape 2018, le programme affiche **En 2018 il y aura 108 adhérents**.
- Si Alain tape 2, le programme affiche le détail des prévisions de 2017 à l'année choisie.
Exemple : si Alain tape 2018 :

En 2017 il y a 100 adhérents.
En 2018 il y aura 108 adhérents.
- Si Alain tape 3, le programme affiche les adhésions cumulées entre 2017 et l'année choisie (chaque adhérent renouvelle chaque année).
Exemple : pour 2019, le programme affiche **De 2017 à 2019 il y aura 325 adhésions cumulées**.
- Si Alain tape 4, le programme affiche **Entrez le nombre d'adhérents voulus** : puis l'année à partir de laquelle ce nombre sera atteint.
Exemple : si Alain tape 115, le programme affiche **On atteindra 115 adhérents en 2019**.
- Si Alain tape Q, le programme affiche **Au revoir**. et s'arrête.
- Si Alain tape autre chose, le programme affiche **Choix invalide, recommencez** : jusqu'à obtention d'un choix valide.

Exemple d'exécution :

Menu, veuillez choisir :

1. Prévisions adhérents à l'année N (résumé)
2. Prévisions adhérents à l'année N (détails)
3. Adhésions cumulées à l'année N
4. Année à laquelle on obtiendra M adhérents

Q. Quitter

4

Entrez le nombre d'adhérents voulus : 203

On atteindra 203 adhérents en 2027.

Menu, veuillez choisir :

1. Prévisions adhérents à l'année N (résumé)

2. Prévisions adhérents à l'année N (détails)

3. Adhésions cumulées à l'année N

4. Année à laquelle on obtiendra M adhérents

Q. Quitter

Q

Au revoir.

Exercice 2- M. Bizarre

M. Bizarre est rédacteur-en-chef d'un journal, et a souvent des idées farfelues. Suivant les jours de la semaine, il souhaite que les articles de son journal suivent des règles qu'il a inventées.

Chaque lundi, il souhaite que les mots soient répétés deux fois, séparés par un espace. Écrire une fonction `lundi` qui prend en argument un mot et qui transforme ce mot selon la règle du lundi. Par exemple, `lundi("bonjour")` doit valoir `"bonjour bonjour"`.

Chaque mardi, il faut que les mots de longueur paire soient répétés 6 fois, en les séparant par un tiret, alors que les mots de longueur impaire doivent être répétés 3 fois, en les séparant par des virgules. Écrire la fonction `mardi` qui prend en argument un mot et qui renvoie la chaîne de caractères suivant la règle du mardi. *Rappel* : la longueur d'une chaîne de caractères est donnée par `len(chaine)`.

Chaque mercredi, il faut que les mots de longueur impaire soient remplacés par le mot `"impair"`. Les mots pairs doivent rester tels qu'ils sont. Écrire la fonction `mercredi` (qui prend en argument un mot et qui renvoie la bonne chaîne de caractères).

Chaque jeudi, il faut que les mots soient répétés autant de fois que leur longueur modulo 3 (à la suite, sans espace). Écrire la fonction `jeudi`. Par exemple, `jeudi("merci")` vaut `"mercimerci"`, `jeudi("bonbon")` vaut `""`, `jeudi("comment")` vaut `"comment"`.

Le vendredi, il faut que les mots soient écrits normalement. Et heureusement, il n'y a pas de journal le week-end.

Écrire une fonction `transforme(mot, num_jour)` qui prend en argument un mot et le numéro du jour (1 pour lundi, 2 pour mardi, etc.) et qui renvoie le mot transformé selon la règle du jour correspondant.

Exercice 3- Le trésor

Vous partez à la recherche d'un trésor caché dans une pyramide. Vous avez réussi à obtenir des informations sur les épreuves qui vous attendent à l'intérieur de celle-ci :

- Dans le hall d'entrée, une statue de sphinx va vous demander un code secret sous la forme suivante :

« Je n'aime que les nombres p qui sont des nombres premiers, tels que $p + 2$ soit aussi un nombre premier... Et mon nombre préféré de la journée est un nombre qui change tous les jours. Le code secret est le plus petit nombre supérieur ou égal à mon nombre préféré parmi les nombres que j'aime. »

- Une fois entré, vous parcourez un labyrinthe dont vous vous êtes déjà procuré le plan, puis vous arrivez face à une statue d'Osiris qui garde la salle du trésor. Il va vous demander un code secret sous la forme suivante :

« Je n'aime que les nombres n tels que $n + 1$ soit une puissance de 2... et il faut également que le reste de la division de n par 5 soit égal à 3... Mon nombre préféré de la journée est un nombre qui change tous les jours. Le code secret est le plus petit nombre supérieur ou égal à mon nombre préféré parmi les nombres que j'aime. »

Pour chacune des deux énigmes, vous n'aurez que 5 secondes pour répondre, sinon la sortie sera verrouillée et vous resterez enfermés dans la pyramide à tout jamais. Pour éviter cela à tout prix, vous décidez d'écrire un programme que vous emporterez sur votre smartphone et qui vous aidera à résoudre les énigmes.

Écrire les fonctions suivantes :

1. `est_premier(n)` — prend en argument un entier ≥ 2 et renvoie `True` si n est premier, `False` sinon.
2. `sphinx_aime(n)` — renvoie `True` si le sphinx aime n , c'est-à-dire si n et $n + 2$ sont premiers.
3. `code_hall(pref)` — renvoie le code secret du hall à partir du nombre préféré du sphinx (entier positif).
4. `est_puissance2(n)` — renvoie `True` si n est une puissance de 2, `False` sinon. (*On ne doit pas utiliser le logarithme, et on cherche une solution efficace.*)
5. `osiris_aime(n)` — renvoie `True` si n est aimé par Osiris, c'est-à-dire si $n + 1$ est une puissance de 2 et $n \bmod 5 = 3$.
6. `code_tresor(pref)` — renvoie le code secret du trésor à partir du nombre préféré d'Osiris.