**Pour chacun des exercices suivants, avant d’écrire l’algorithme recenser les variables dont vous aurez besoin et précisez leur type.**

Exercice 1 :

1) Écrire un programme qui calcule le volume d’un parallélépipède rectangle après avoir demandé à l’utilisateur la longueur, la largeur, la hauteur.

Exercice 2 :

1) Écrire un programme qui demande à l’utilisateur deux valeurs et affiche la plus grande des deux. Écrit une liste de tests pertinents.

2) Écrire un programme qui demande à l’utilisateur trois valeurs et affiche la plus grande des trois.

Exercice 3 :

1) Écrire un programme qui écrit la table de multiplication du 2 sous la forme :

0\*2=0

1\*2=2

2\*2=4

etc …

2) Écrire un programme qui écrit la table de multiplication d’un nombre entré par l’utilisateur.

3) Écrire un programme qui permet d’écrire toutes les tables de multiplication de 1 à 10.

Exercice 4 :

1) Écrire un algorithme qui recueille au clavier les températures de 7 jours consécutifs et calcule la température moyenne de la semaine.

2) Essaye d’optimiser cet algorithme afin d’utiliser le moins de variables possibles.

Exercice 5 :

On souhaite écrire un algorithme qui permet de vérifier si une date saisie par un utilisateur est valide ou non

1) Sous quelle forme demander la date à l’utilisateur ?

2) Quelles sont les différentes conditions pour qu’une date soit valide ?

3) Écrire le programme correspondant et écris une liste de tests pertinents.

Exercice 6 :

1) Ouvre le fichier *recup\_date.py* et analyse-le. On y explique comment récupérer la date du jour.

2) Écrire l’algorithme qui permet de calcul l’âge d’un utilisateur après lui avoir demandé sa date de naissance.

Exercice 7 :

1. Écrire un programme nommé *JeuHasard* dont l’objectif est de retrouver en un nombre minimal d’essai un nombre entier tiré au hasard entre 1 et 1 000.

Il se décompose selon les étapes suivantes :

• Tirage au hasard d’un entier entre 1 et 1000. Pour cela on importera la bibliothèque **random** en écrivant au début du code **from random import\***.

On utilisera alors la fonction **randrange(N)** qui renvoie un nombre entier compris entre 0 et N-1.

• Proposition d’un entier par l’utilisateur et, selon la valeur proposée, un des messages suivants est communiqué :

- Trop grand !

- Trop petit !

- Bravo, vous avez trouvé en *k* essais.

• L’étape précédente est répétée tant que l’utilisateur n’a pas trouvé et que le nombre d’essais est inférieur à 20. Si le nombre d’essais atteint 20 alors le message suivant apparaîtra : Perdu ! Le nombre à découvrir était …

2) Améliorations possibles.

a. Complète l’algorithme précédent en proposant à l’utilisateur de faire une nouvelle partie s’il le désire.

b. Compte le nombre de parties jouées et le nombre de parties gagnées.

Exercice 8 :

On veut créer un exercice de révisions des tables de multiplications.

On demande à l’élève de choisir un nombre d’opérations souhaité (10 ou 20) puis les tables sont sélectionnées aléatoirement.

Pour chaque opération, si l’élève répond bien la première fois, il gagne 2 points. S’il répond faux, il a droit à un deuxième essai. Si cette fois la réponse est correcte il gagne 1 point. Si la réponse est encore fausse il ne gagne rien et on lui indique la bonne réponse.

À la fin de la série, on lui donnera le nombre de bonnes réponses et le score ramené sur 100.

1) Écrit l’algorithme correspondant.

2) On souhaite maintenant pouvoir lui proposer de recommencer une série pour éventuellement améliorer son score.

À la fin de la nouvelle série, le nombre de bonnes réponses et le score de la série sont de nouveau affiché ainsi que le meilleur score effectué sur l’ensemble des séries.

Modifie l’algorithme précédent.