Algorithmique et programmation 1
Feuille d'exercices 3 - Boucles

Exercice 1 — Premières boucles

1. Écrire un algorithme, contenant une boucle **Pour**, qui demande à l'utilisateur un entier `nb` puis qui affiche la table de multiplication par `nb`, en affichant les multiples de `nb` comme à l'école. Si l'utilisateur saisit la valeur 23, on obtiendra un affichage du genre :

```
23 * 1 = 23
23 * 2 = 46
[... votre algo doit tous les afficher...]
23 * 10 = 230
```

2. Que doit-on changer dans cet algorihtme pour remplacer la boucle **Pour** par une boucle **Tant que** ?

Exercice 2 — Devine !

On suppose que l'on dispose d'une fonction aleaentier qui génère des entiers aléatoirement. Par exemple aleaentier(3,9) génère un entier qui est supérieur ou égal à 3 et strictement inférieur à 9. Écrire un algorithme qui génère un entier compris entre 0 et 99 puis demande à l'utilisateur de rentrer un entier tant qu'il n'a pas trouvé la bonne valeur.

A chaque essai on affiche un commentaire :
— si l'utilisateur a trouvé on lui dit bravo et on lui donne le nombre d'essais réalisés pour gagner,
— si la valeur entrée est trop petite, on le signale pour aider le joueur,
— si la valeur est trop grande, on le signale aussi.

Exercice 3 — Stop ou encore ?

Cet exercice consiste à écrire un algorithme qui simule un jeu de dés. Le jeu commence par lancer deux dés et définir une valeur (la somme des deux dés) à éviter à tout prix. Le joueur va ensuite lancer les deux dés. Si la somme est différente de celle à éviter, il ajoute la valeur obtenue à son propre score. Si elle est égale à la valeur à éviter, le joueur a perdu. À chaque lancer on affiche le score et on demande au joueur s'il veut continuer ou non.

Exemples :

Résultat du lancer : 2 et 5
Somme à éviter : 7
Résultat du lancer : 3 et 3
Score : 6
Voulez-vous rejouer (oui/non) ? oui
Résultat du lancer : 6 et 1
Vous avez perdu !

Résultat du lancer : 1 et 5
Somme à éviter : 6
Résultat du lancer : 3 et 6
Score : 9
Voulez-vous rejouer (oui/non) ? oui
Résultat du lancer : 6 et 1
Score : 16
Voulez-vous rejouer (oui/non) ? non
Vous avez marqué 16 points.

Exercice 4 — Présentation d'un algorithme

Présentez correctement l'algorithme ci-après puis devinez ce qu'il fait.

Algorithme mystère **Variables** m, debinterv, fininterv, y : réel trouve : booléen **Début** debinterv ← 0.1 fininterv ← 20 trouve ←**Faux Tant que** (trouve = **Faux**) **Faire** m ← (debinterv+fininterv) / 2 y ← ln(m) **Si** (abs(y-1) < 0.0001) **Alors** trouve ←**VraiSinon Si** (y > 1) **Alors** fininterv ← m **Sinon** debinterv ← m **Finsi Finsi Fintantque** afficher ("On trouve ", m) **Fin**

Exercice 5 — PGCD

Le PGCD de deux nombres peut se calculer en remplaçant le plus grand par son écart au plus petit, et ce jusqu'à obtention de deux nombres égaux.

Exemple :
PGCD (110,154) = PGCD (110,44) = PGCD (66,44) = PGCD (22,44) = PGCD (22,22) = 22.

1. Écrire une fonction qui prend en paramètres deux entiers (supposés positifs) et retourne leur PGCD.
2. Écrire un algorithme qui demande deux entiers à l'utilisateur, recommence tant qu'un des deux est nul, puis affiche leur PGCD (en appelant la fonction).

Exemple d'exécution :
Entrez deux entiers : 0
154
Les deux entiers doivent être non nuls
Entrez deux entiers : 110
154
Le PGCD de 110 et 154 vaut : 22

Exercice 6 — Plus rond, tu meurs

Écrire un algorithme qui demande un réel positif à l'utilisateur puis propose répétitivement le menu suivant jusqu'à ce que l'utilisateur réponde 0, pour quitter :

> Entrer un entier en fonction de l'opération que vous voulez effectuer :
> 1 : Entrer un nouveau rayon.
> 2 : Calcul du périmètre du cercle.
> 3 : Calcul de la surface du cercle.
> 4 : Calcul de l'aire de la sphère.
> 5 : Calcul du volume de la sphère.
> 0 : Sortir du programme.
> Votre choix ?

Si l'utilisateur entre un rayon négatif, il faut le lui signaler et lui en demander un nouveau. Si par contre il entre par exemple un rayon égal à 2,5 puis choisit l'option 2 du menu, l'algorithme affiche le résultat du calcul sous la forme suivante :

"Le périmètre du cercle de rayon 2,5 est : 15,707963267"

Rappel : l'aire et le volume d'une sphère sont respectivement $4.\pi.R^2$ et $4/3.\pi.R^3$.

Exercice 7 — Calculs de π

1. Écrire une fonction en pseudo-langage `calcpi` qui prend en paramètre un entier k et retourne une valeur approchée de π à l'aide de la fonction zêta de Riemann en allant sommant les k premiers termes :

$$\zeta(2) = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + ... + \frac{1}{k^2} + ... = \frac{\pi^2}{6}$$

2. Ecrire une nouvelle fonction `calcpi2` qui prend comme paramètre un réel s et calcule (et somme) les termes de la série ci-dessus jusqu'à ce que le dernier terme calculé soit inférieur au seuil s.
3. Reprendre la question ci-dessus pour créer une fonction `calcpi3` se basant sur la formule suivante :

$$\pi = 4 \times (1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + ...)$$

4. Ecrire un fragment d'algorithme qui appelle (et affiche le résultat de) votre première fonction `calcpi` avec comme paramètre 1 puis 2 et ainsi de suite jusque 10
5. Le compléter pour qu'il demande un seuil à l'utilisateur et affiche le résultat de vos deuxième et troisième fonctions sur ce seuil.

Exercice 8 — Bob le robot

Bob le robot est positionné sur un damier de dimensions entières suffisamment grandes pour qu'on ne s'occupe pas des bords. Il est commandé par l'utilisateur qui lui indique une direction (N,O,S,E) et un nombre de pas.

1. Le robot étant initialement positionné sur la case de coordonnées(0,0), écrire un algorithme qui demande à l'utilisateur une direction et un nombre de pas et calcule puis affiche la position du robot à l'issue de ce déplacement.
2. Compléter l'algorithme précédent pour faire calculer la position du robot au bout de N opérations du type précédent (N demandé à l'utilisateur).

Exercice 9 — Nombres parfaits

Un nombre entier est parfait si et seulement si il est égal à la somme de ses diviseurs autres que lui-même.

Exemple : 6 (= 1 + 2 + 3) et 28 (= 1 + 2 + 4 + 7 + 14) sont parfaits

1. Écrire une fonction qui détermine si un entier est parfait.
2. Écrire un algorithme qui demande un entier à l'utilisateur, vérifie qu'il est supérieur ou égal à 2, puis, si c'est le cas, affiche tous les nombres parfaits entre 2 et ce nombre saisi. Si l'entier est strictement inférieur à 2 on affichera un message d'erreur.

Exercice 10 — Pour ceux qui auraient tout fini

On propose la variante suivante à l'algorithme du robot : un pirate mal intentionné a piégé deux cases du damier, mais on ne sait pas lesquelles. La position des pièges (xp1,yp1) et (xp2,yp2) est déterminée aléatoirement en début d'algorithme. Écrire l'algorithme correspondant à cette variante : le processus s'arrête soit au bout de N déplacements, soit quand le robot tombe dans un des deux pièges ; l'algorithme affiche alors le nombre de déplacements effectués avant la chute.