

## TP : Algorithmes utilisant les boucles

Buts : Boucles.

Le but de ce TP est de manipuler les boucles dans différents exercices indépendants les uns des autres.

### Exercice 1 : Pliage d'une feuille

Une feuille de papier ordinaire mesure environ 0,16mm d'épaisseur. On décide de la plier en deux, puis en 4, puis en 8, etc.

Combien faut-il de pliages pour arriver à une épaisseur supérieure à la hauteur de la Tour Eiffel (324m) ?

Écrire un algorithme (sur papier) qui permet de répondre à la question.

Implémenter cet algorithme en C++.

### Exercice 2 : compte à rebours

Écrire un algorithme (sur papier) demandant à l'utilisateur de saisir un nombre entier positif  $n$  et qui affiche tous les nombres entiers dans l'ordre décroissant de  $n$  à 0.

Implémenter cet algorithme en C++.

### Exercice 3 : Jeu des petits chevaux

Le jeu des petits chevaux est un jeu de société. Pour gagner il faut être le premier à amener ses pions (des chevaux) à la fin du parcours. Le jeu se joue avec un dé à 6 faces. Pour pouvoir sortir un cheval de son écurie, le joueur doit faire un 6.

Écrire un algorithme (sur papier) qui simule les tirages successifs du dé jusqu'à ce que le cheval puisse sortir de l'écurie.

Implémenter cet algorithme en C++.

*Remarque* : la fonction qui renvoie un nombre aléatoire est `rand()`. Elle renvoie un entier positif aléatoire. Pour ramener cet entier entre 1 et 6, on peut utiliser l'opérateur modulo `%` de cette façon : `rand()%6 + 1` (l'opérateur modulo `%` ramène le nombre entre 0 et 5, on rajoute donc 1 pour avoir un nombre entier 1 et 6). Mais si vous essayez de lancer plusieurs fois la fonction, vous vous apercevez qu'elle vous renvoie toujours la même suite de nombres. C'est parce qu'il n'existe aucun aléa dans un ordinateur, il est entièrement déterministe. On va alors faire du pseudo-aléatoire en demandant à la fonction `rand` de choisir une autre liste de nombres en fonction de l'heure qu'il est en déclarant au préalable :

```
srand (time(NULL));  
rand()%6 + 1;
```

Il faudra rajouter les bibliothèques suivantes :

```
#include <ctime> //pour la fonction time  
#include <cstdlib> // pour la fonction rand
```

### Exercice 4 : mot de passe

Pour accéder à un forum, un étudiant doit se connecter à l'aide d'un mot de passe. Tant qu'il ne saisit pas le bon mot de passe, l'ordinateur lui demande de le saisir à nouveau.

- Écrire un algorithme (sur papier) qui demande à l'étudiant son mot de passe autant de fois que nécessaire pour accéder au forum. Implémenter cet algorithme en C++.

*Remarque :* les chaînes de caractères sont gérées par le type `string` en C++. Pour pouvoir les utiliser, il faut inclure la bibliothèque `cstring` en ajoutant la ligne suivant avant le `main()` :

```
#include <cstring>
```

Pour déclarer une chaîne de caractères, écrire :

```
string machaine;
```

- Modifier l'algorithme précédent pour que l'étudiant soit limité à trois propositions. Pour cela on peut mettre en place un compteur.