

**TD 3 : Algorithmes : instructions conditionnelles, boucles simples**INSTRUCTIONS CONDITIONNELLES : **IF****Exercice 1.**

- (1) Écrire un programme qui demande à l'utilisateur de rentrer son âge puis affiche un message indiquant s'il est majeur ou non.
- (2) Modifier le programme précédent pour qu'il indique si l'utilisateur a droit à une réduction SNCF : **"enfant"** (moins de 11 ans), **"jeune"** (12-27 ans) ou **"senior"** (60 ans ou plus) ; sinon afficher **"plein tarif"**.

**Exercice 2.**

- (1) Les magasins sont généralement ouverts de 10h à 12h et de 14h à 19h. Écrire un programme qui demande à l'utilisateur l'heure et lui indique si les magasins sont ouverts ou non.
- (2) Si nécessaire, modifier votre programme précédent pour qu'il n'y ait qu'un seul **if**.

INSTRUCTIONS ITÉRATIVES : BOUCLE **WHILE****Exercice 3.**

Écrire un programme qui demande à l'utilisateur un entier négatif. Tant que le nombre entré est positif ou nul, le programme continue de demander un nombre à l'utilisateur.

**Exercice 4.**

Écrire un programme pour le jeu « Plus ou Moins » dont le but est de deviner un nombre choisi aléatoirement entre 0 et 1000 par l'ordinateur (on pourra supposer que l'expression `aleaint()` donne un tel nombre aléatoire).

À chaque étape et jusqu'à ce qu'il trouve le bon nombre, le joueur propose un nombre à l'ordinateur qui lui indique si ce nombre est inférieur ou supérieur au nombre à trouver. Une fois que le joueur a trouvé le bon nombre, on affiche le nombre d'essais du joueur.

INSTRUCTIONS ITÉRATIVES AVEC COMPTEUR : BOUCLES **WHILE** ET **FOR****Exercice 5.**

Pour chaque point suivant, écrire un programme qui demande à l'utilisateur un entier  $n$  puis

- (1) affiche les nombres de 1 à  $n$  ;
- (2) affiche les nombres de  $n$  à 1 ;
- (3) affiche les nombres entiers de carré inférieur à  $n$  ;
- (4) affiche la valeur de la somme  $1^2 + 2^2 + \dots + n^2$ .

**Exercice 6** (boucles **for**).

Reprendre les deux exercices précédents en utilisant des boucles **for** ; à votre avis, dans chacun des cas, laquelle des deux formes est la plus naturelle ?

**Exercice 7.**

Deviner l'affichage du fragment de programme mystère suivant :

```
for ( int i = 1; i <= 4; i++ ) {  
    for ( int j = 1; j <= i; j++ ) {  
        cout << "(" << i << "," << j << ") ";  
    }  
    cout << endl;  
}
```

**POUR ALLER PLUS LOIN****Exercice ♣ 8** (Nombres premiers).

- (1) Écrire un programme qui demande un entier à l'utilisateur, puis répond si oui ou non cet entier est un nombre premier.
- (2) Écrire un programme qui demande un entier  $n$  à l'utilisateur, puis affiche tous les nombres premiers entre 1 et  $n$ .
- (3) Écrire un programme qui affiche les  $n$  premiers nombres premiers.

**Exercice ♣ 9** (Dates).

- (1) Écrire un programme qui demande à l'utilisateur une date sous la forme de trois entiers jour / mois / année, et réponde si oui ou non c'est une date possible. Pour l'instant, on ignore les années bissextiles.

Exemple d'exécution :

```
Entrez une date: 28 5 1973  
Cette date est valide  
  
Entrez une date: 31 2 2015  
Cette date n'est pas valide
```

- (2) Faire un programme qui demande à l'utilisateur de rentrer une date et, si la date est valide, affiche la date du lendemain.

Exemple d'exécution :

```
Entrez une date: 14 12 2015  
Le jour suivant est le 15 12 2015  
  
Entrez une date: 31 5 1859  
Le jour suivant est le 1 6 1859
```

- (3) Les dates de ces exemples ont-elles été choisies au hasard ?
- (4) Reprendre la question 1 en considérant les années bissextiles (une année est bissextile si elle est divisible par 4, mais pas 100 sauf si elle est divisible par 400).
- (5) Écrire un programme qui demande une date à l'utilisateur, puis donne le jour de la semaine de cette date.

- (6) Projet Euler- 19 : Combien de premiers du mois ont été des dimanches au XX<sup>e</sup> siècle ?