

TD 9

Concevoir des fonctions

R1.01 - Initiation au développement

Exercice 1 : Valeur absolue

- a/ Écrivez la **déclaration** d'une fonction qui retourne la valeur absolue d'un nombre.
- b/ Écrivez un exemple d'**appel** de cette fonction pour afficher la valeur absolue d'un nombre *nb* précédemment saisi.
- c/ Écrivez la **définition** de cette fonction.

Exercice 2 : Vérification d'une note

- a/ Écrivez la **déclaration** d'une fonction qui détermine si un nombre est une note (valeur comprise entre 0 et 20) ou pas.
- b/ Écrivez un **exemple** d'appel de cette fonction pour afficher si un nombre *nb* précédemment saisi est une note ou pas.
- c/ Écrivez la **définition** de cette fonction.

Exercice 3 : Adresse d'une personne

On considère les types *Adresse* et *Personne* tels que définis sur la page suivante.

- a/ Écrivez la **déclaration** d'une fonction qui retourne l'adresse d'une personne donnée.
- b/ Écrivez un exemple d'**appel** de cette fonction.
- c/ Écrivez la **définition** de cette fonction.

```
1 struct Adresse
2 {
3     string numRue; // par exemple "2bis"
4     string nomRue; // par exemple "allée du Parc Montauray"
5     unsigned int codePostal; // par exemple 64600
6     string ville; // par exemple "Anglet"
7 };
8
9 struct Personne
10 {
11     string nom; // Le nom de la personne
12     string prenom; // Le prénom de la personne
13     Adresse adresse; // L'adresse de la personne
14 };
```

Exercice 4 : Moyenne des valeurs d'un tableau de nombres

- a/ Écrivez la déclaration d'une fonction qui détermine la moyenne des valeurs stockées dans un tableau.
- b/ Écrivez un exemple d'appel de cette fonction pour afficher la moyenne des valeurs contenues dans un tableau *notes* de taille *nbCases*.
- c/ Écrivez l'algorithme de cette fonction.

Exercice 5 : Nombre d'employés habitant une ville donnée

On considère les types *Adresse* et *Personne* tels que définis dans l'exercice 3. On suppose également qu'on dispose d'un tableau contenant tous les employés de l'IUT de Bayonne et défini de la manière suivante :

```
1 Personne employesIut [95]; // Liste des employés de l'IUT
```

- a/ Écrivez la déclaration d'une fonction qui retourne l'adresse d'une personne donnée.
- b/ Écrivez un exemple d'appel de cette fonction.
- c/ Écrivez la définition de cette fonction.

Exercice 6 : Le jeu du lièvre et de la tortue

On souhaite mettre en place un programme qui simule la course entre le lièvre et la tortue (en référence à la fable de La Fontaine) selon les règles suivantes : A chaque

tour, on lance un dé. Si le 6 sort, le lièvre gagne la partie, sinon la tortue avance d'une case. La tortue gagne quand elle a avancé de six cases.

L'objectif final est de déterminer si le jeu, tel qu'il est défini, est à l'avantage du lièvre ou de la tortue...

a/ Écrivez la déclaration d'un sous-programme qui simule **une** course et indique qui est le vainqueur.

b/ Écrivez l'algorithme de ce sous-programme en supposant que vous disposez d'une fonction *random()* répondant aux spécifications suivantes :

```
1 int random (int min, int max);
2 // renvoie un entier aléatoire sur l'intervalle [min,max]
```

c/ Élaborez à présent un algorithme qui simule un nombre de courses *nbCourses* et qui calcule le nombre de parties gagnées par le lièvre et le nombre de parties gagnées par la tortue. En TP, vous utiliserez cet algorithme pour coder un programme qui réponde à la question initiale...

Exercice 7 : Calendrier perpétuel

L'algorithme de Zeller permet de déterminer le jour de la semaine (lundi, mardi...) d'une date donnée.

Soit une date exprimée sous la forme (*jour, mois, année*).

Parce que le jour rajouté à une année bissextile l'est fin février, les calculs sont facilités si on fait commencer l'année le 1er mars. Pour cela, à partir de (*jour, mois, année*), on calcule une date modifiée (*jour, decalageMois, decalageAnnee*) selon les exemples ci-dessous :

jour	mois	annee	→	jour	decalageMois	decalageAnnee
1	1	1994	devient	1	11	1993
1	2	1994	devient	1	12	1993
1	3	1994	devient	1	1	1994
1	4	1994	devient	1	2	1994
1	5	1994	devient	1	3	1994
1	6	1994	devient	1	4	1994
1	7	1994	devient	1	5	1994

jour	mois	annee	→	jour	decalageMois	decalageAnnee
1	8	1994	devient	1	6	1994
1	9	1994	devient	1	7	1994
1	10	1994	devient	1	8	1994
1	11	1994	devient	1	9	1994
1	12	1994	devient	1	10	1994

La suite de l'algorithme consiste à découper *decalageAnnee* en deux parties distinctes *decalageAnneePartie1* et *decalageAnneePartie2* de la manière suivante :

Si *decalageAnnee* = 1994, alors *decalageAnneePartie1* = 19 et *decalageAnneePartie2* = 94.

Pour terminer, on applique la formule :

$$\text{numJour} = (\text{jour} + (13 \times \text{decalageMois} - 1) / 5 + \text{decalageAnneePartie2} + \text{decalageAnneePartie2} / 4 + \text{decalageAnneePartie1} / 4 + 5 \times \text{decalageAnneePartie1}) \text{ modulo } 7$$

et on déduit le jour correspondant de la manière suivante : numJour = 0 → dimanche, numJour = 1 → lundi, numJour = 2 → mardi, . . . , numJour = 6 → samedi.

a/ Écrivez la **déclaration** d'une fonction qui détermine le jour d'une date donnée via l'algorithme de Zeller.

b/ Écrivez un exemple d'**appel** de cette fonction pour afficher le jour de votre naissance.

c/ Écrivez l'algorithme de cette fonction.

Après avoir codé cette fonction en TP, vérifiez que le 15 octobre 1582 était un vendredi. Cherchez ensuite votre jour de naissance.