

LIF 1 – Mercredi 9 novembre 2011

Durée 1h

Les documents, calculatrice, téléphone portable sont interdits

La qualité de l'écriture et de la présentation sera prise en compte dans la note finale.

Vous veillerez à respecter les notations et les règles d'écriture des algorithmes vues en cours et en TD.

Soyez attentifs aux indices des tableaux et prenez garde à ne pas dépasser la fin des tableaux.

Le barème est donné à titre indicatif.

Partie A : Questions de cours (4 points)

- 1- Donner le codage binaire de la valeur 97 sur 1 octet. (1.5 pts)
- 2- Ecrire l'instruction C/C++ permettant de transformer la constante caractère '4' en sa valeur numérique 4. On pourra stocker le résultat dans une variable appelée `quatre` de type entier. (1.5 pt)
- 3- Ecrire des deux manières possibles différentes en langage C/C++ la déclaration d'une constante MAX qui vaut 100. (1 pt)

Partie B : Algorithmique (8 points)

- 1- **Les nombres parfaits** (5 pts) : Un nombre parfait est un entier naturel n non nul qui est égal à la somme de ses diviseurs stricts (n exclus). Exemple : $6 = 1 + 2 + 3$
 - a. Écrire en langage algorithmique un sous-programme qui retourne vrai si un entier n passé en paramètre est un nombre parfait, faux sinon. (3 pts)
 - b. Écrire en langage algorithmique le programme principal permettant d'afficher les nombres parfaits compris entre 1 et 10000. (2 pts)
- 2- **Distance à la diagonale** (3 pts) : Écrire en langage algorithmique un sous-programme permettant de remplir un tableau 2D carré (matrice) de taille 5 de la manière suivante :

- 0 sur la diagonale
- 1 sur les lignes au-dessus et en dessous de la diagonale
- ...
- n sur la $n^{\text{ième}}$ ligne au-dessus et en dessous de la diagonale

0	1	2	3	4
1	0	1	2	3
2	1	0	1	2
3	2	1	0	1
4	3	2	1	0

Attention : la solution proposée ne doit pas comporter plus d'une dizaine de lignes.

Partie C : Langage C (8 points)

- 1- Ecrire en langage C/C++ une **fonction** `saisie_positive` permettant de saisir et de retourner une valeur strictement positive. La fonction devra recommencer la saisie jusqu'à ce que la valeur soit strictement positive. (2 pts)
- 2- Ecrire en langage C/C++ une **fonction** booléenne `est_diviseur` qui renverra vrai si b divise a et faux sinon. (2 pts)
Exemples : `est_diviseur (6, 3)` renverra vrai et `est_diviseur (5, 2)` renverra faux
- 3- Ecrire en langage C/C++ une **procédure** `diviseurs_2_3_5` permettant de tester si un entier strictement positif passé en paramètre est divisible par 2, par 3, par 5. Cette procédure renverra 3 résultats booléens `div2`, `div3` et `div5`. (2 pts)
Par exemple : 10 est divisible par 2 et 5 donc `div2 = true`, `div3 = false` et `div5 = true`.
- 4- Ecrire en langage C/C++ la fonction principale permettant de saisir un nombre strictement positif et d'afficher s'il est divisible par 2, par 3, et/ou par 5. (2 pts)