

# INFORMATIQUE

Semestre 1

|                        |  |
|------------------------|--|
| Nom de l'U.E. :        | <b>Initiation à l'algorithmatique et aux outils informatiques usuels</b> |
| Code de l'U.E. :       | <b>X2I0040</b>   |
| Date de l'examen :     | 12 mai 2015  |
| Durée :                | 1h20   |
| Documents autorisés :  | aucun  |
| Calculatrice autorisée | non  |

### Présentation et syntaxe des algorithmes (1 point)

Pour valider ce point, il est demandé de commenter proprement vos algorithmes, ainsi que d'utiliser les mêmes notations que celles utilisées en cours.

### Questions à choix multiple sur les cours (réponse correct : 0.5, réponse fausse : -0.5)

```

fonction double(x)
variable d: entier
début
    d ← 2*x
    retourner d
fin

```

```

fonction fact(n : entier) : entier
variables
res : entier
début
    si (n=0)
        alors res ← 1;
    Sinon
        _____ ;
    fin si
    retourner(res);
fin

```

E. Si le système dynamique ci-contre s'applique à deux espèces (x et y), quelle est la signification de beta ?

- digestion     assimilation     prédition     reproduction

F. Définir un ensemble en extension, c'est

- en donner explicitement tous les éléments  
 donner une formule permettant de vérifier si un élément est dans l'ensemble ou pas  
 allonger l'ensemble de façon à ce qu'il soit ovale

G. Dans un réseau social, on appelle le fait que les individus ne sont pas très distants les uns des autres

- l'effet de la distance logarithmique  
 l'effet du petit monde  
 le phénomène de la distance logarithmique

A. Dans la fonction ci-contre x est

un argument d'appel  
 un argument formel  
 un argument de déclaration

B. Le corps de la fonction double est

la fonction elle-même  
 l'endroit où elle est appelée  
 les lignes 2 à 5 de la fonction

C. Quelle est la ligne manquante de la fonction ci-contre

- res ← n\*fact(n-1);  
 res ← res\*n;  
 res ← n\*n-1;  
 res ← n\*(n-1);

D. Comment appelle-t-on cette fonction

- successive  
 accumulative  
 laxative  
 recursive

$$\begin{cases} \frac{dx(t)}{dt} = x(t) (\alpha - \beta y(t)) \\ \frac{dy(t)}{dt} = -y(t) (\delta - \gamma x(t)) \end{cases}$$

H. Dans une conditionnelle, la condition

- peut être un entier  
 peut être n'importe quelle expression booléenne  
 peut-être uniquement une variable booléenne

I. Diffie-Hellman désigne

- un générateur aléatoire  
 un cryptosystème symétrique  
 un cryptosystème asymétrique  
 un protocole d'échange de clés

J. Lequel de ces termes NE DESIGNE PAS un paradigme de programmation

- Programmation impérative
- Programmation fonctionnelle
- Programmation factorielle
- Programmation évènementielle

### Exercice 1 (CM semaine 10) : 2 points

Ecrire l'algorithme d'une fonction de similarité qui pour deux caractères passés en arguments, et retourne 1 en cas de similarité et 0 en cas de non similarité.

### Exercice 2 (CM semaine 11) : 3 points

Ecrire un algorithme qui transforme une matrice de similarité de séquence en une matrice de distance. Vous utiliserez pour cela la variable suivante `M_sim` Tableau de taille  $n \times n$ : entiers

### Exercice 3 (CM semaine 13) : 4 points

Pour l'étude de vers vestimentaires hydrothermaux *Riftia pachyptila*, la taille de 150 individus a été mesurée et stockée dans `X` Tableau de Taille 150: réels. La population peut être rangée en 3 classes de taille. Les classes sont délimitées par deux seuils (`seuil_a` et `seuil_b`). Ecrire un algorithme qui compte le nombre d'individus par classe de taille et qui stocke ces valeurs dans `Nb_tab` Tableau de Taille 3: entiers

### Exercice 4 : 5 points

Les différentes questions, bien que liées, peuvent être traitées de manière indépendante. Le triangle de Pascal est un moyen pratique pour connaître les coefficients de  $(x + y)^n$ . Un triangle de Pascal est obtenu de la manière suivante :

- 1 sur la première colonne
- Chaque ligne suivante commence et termine par 1, chaque élément non extrême est égal à la somme des deux éléments de la ligne précédente qui se trouvent sur la colonne précédente et sur la colonne de l'élément calculé.

Dans un algorithme, déclarer un tableau d'entiers nommé `triangle` à 2 dimensions de taille 10 x 10.

Par ailleurs, `triangle[i][j]` permet d'accéder à la valeur de `triangle` à la ligne `i` et à la colonne `j`.

1) Créer un algorithme qui initialise un tableau `triangle` en mettant la valeur 0 dans chacune des cases.

2) Compléter l'initialisation avec des instructions permettant de mettre la valeur 1 dans la première colonne.

3) Supposant le tableau correctement initialisé, affecter les valeurs comme dans l'exemple ci-dessus. Pour cela, on effectuera le remplissage des valeurs suivant les règles exposées plus haut. On remarquera que la dernière valeur de la ligne `i`, est en colonne `i`

|   |   |    |    |     |     |    |    |   |   |   |
|---|---|----|----|-----|-----|----|----|---|---|---|
| 1 | 0 | 0  | 0  | 0   | 0   | 0  | 0  | 0 | 0 | 0 |
| 1 | 1 | 0  | 0  | 0   | 0   | 0  | 0  | 0 | 0 | 0 |
| 1 | 2 | 1  | 0  | 0   | 0   | 0  | 0  | 0 | 0 | 0 |
| 1 | 3 | 3  | 1  | 0   | 0   | 0  | 0  | 0 | 0 | 0 |
| 1 | 4 | 6  | 4  | 1   | 0   | 0  | 0  | 0 | 0 | 0 |
| 1 | 5 | 10 | 10 | 5   | 1   | 0  | 0  | 0 | 0 | 0 |
| 1 | 6 | 15 | 20 | 15  | 6   | 1  | 0  | 0 | 0 | 0 |
| 1 | 7 | 21 | 35 | 35  | 21  | 7  | 1  | 0 | 0 | 0 |
| 1 | 8 | 28 | 56 | 70  | 56  | 28 | 8  | 1 | 0 | 0 |
| 1 | 9 | 36 | 84 | 126 | 126 | 84 | 36 | 9 | 1 | 0 |