INFORMATIQUE

Semestre 2

Licence 1
X2I0010 Algorithmique et programmation

Université de Nantes
2013/2014

Examen

Aucun document autorisé. Calculatrice non autorisée.
Durée : 1h30

Exercice 1 (J'ai la mémoire qui flanche, barème 4 pts)

1. Effectuer une simulation tabulaire de l'exécution du programme ci-dessous.

```
type
   t_point = enregistrement
      entier abs
      entier ord
   fin enregistrement

variables
   entier i, j
   pointeur vers entier pi
   pointeur vers t_point ptrp
debut
1  i ← 3
2  pi ← adresse(i)
3  memoire(pi) ← 1
4  pi ← allocation(entier)
5  memoire(pi) ← 6
6  ptrp ← allocation(t_point)
7  memoire(ptrp).abs ← i
8  memoire(ptrp).ord ← memoire(pi) + 4
9  pour j de 0 a i faire
10  ecrire ("j = ", j)
11 desallouer(ptrp)
12 fin
```

2. Énoncer l'état de la mémoire à la fin de l'exécution.

Exercice 2 (Tableau , barème 4 pts)

Il s'agit d'écrire un sous-algorithme qui détermine la valeur minimale d'un tableau d'entiers de taille N dont les nb premières cases sont occupées.

1. Définir le type nécessaire pour la déclaration du tableau.
2. Écrire une version itérative du sous-algorithme.
3. Écrire une version récursive du sous-algorithme.

Exercice 3 (Un train peut en cacher un autre, barème 12 pts)

Un billet de train (simplifié) est composé d'un horaire et d'une ville de départ, d'un horaire et d'une ville d'arrivée. En sus, est indiqué le prix du voyage.
Pour cet exercice, on suppose qu'aucun train ne circule entre 23h59 et 0h01.

1. Définir le type *t_temps* qui regroupe les informations temporelles (heures et minutes uniquement).
   Exemple : `17h31`
2. Écrire un sous-algorithme qui force l'utilisateur à saisir un horaire valide : le nombre d'heures est inférieur à 24 et le nombre de minutes inférieur à 60.
3. Écrire une fonction booléenne qui prend en entrée deux paramètres de type *t_temps* nommés $horaire1$ et $horaire2$ et qui contrôle si $horaire1$ précède $horaire2$.
   Exemple : `17h31` précède `18h48`
4. Définir le type *t_billet* qui regroupe les informations concernant un billet de train comme indiqué dans l'énoncé.
5. Écrire un sous-algorithme qui permet de saisir un billet de train valide pour lequel l'heure de départ précède l'heure d'arrivée.
6. Écrire un sous-algorithme qui affiche un billet de train.
7. Écrire un sous-algorithme qui calcule la durée d'un voyage en train.
8. Écrire un algorithme qui permet à l'utilisateur de saisir un billet de train, puis qui affiche ce billet ainsi que la durée du voyage.

UNIVERSITÉ DE NANTES

U.F.R. des Sciences et des Techniques

Année universitaire 2013-2014
Session 1

Nom de l'U.E. :	**Initiation à l'algorithmique et aux outils informatiques usuels**
Code de l'U.E. :	**X2I0040**
Date de l'examen :	14/05/2013
Durée :	1h30
Documents autorisés :	aucun
Calculatrice autorisée	non

Exercice 1 (4 points)

A) Saisie()
1. est une fonction à 0 arguments
2. n'est pas une fonction
3. est une phonction

B) `V vecteur [1..100] d'entiers`
On suppose V rempli et i compris entre 1 et 100. On n'a pas le droit d'écrire
1. V⟵V+1
2. V[i]⟵i+1
3. V[i+1]⟵i

C) Définir un ensemble en extension, c'est
1. en donner explicitement tous les éléments
2. donner une formule permettant de vérifier si un élément est dans l'ensemble ou pas
3. allonger l'ensemble de façon à ce qu'il soit ovale

D) Le seul cryptosystème prouvé inconditionnellement sûr est:
1. le codage de César
2. le codage de Vernam
3. Le code Rebecca.
4. Le codage de Vigénère.

E) La loi de Zipf classe les mots
1. selon leur fréquence
2. selon leur première appararition
3. alphabétiquement

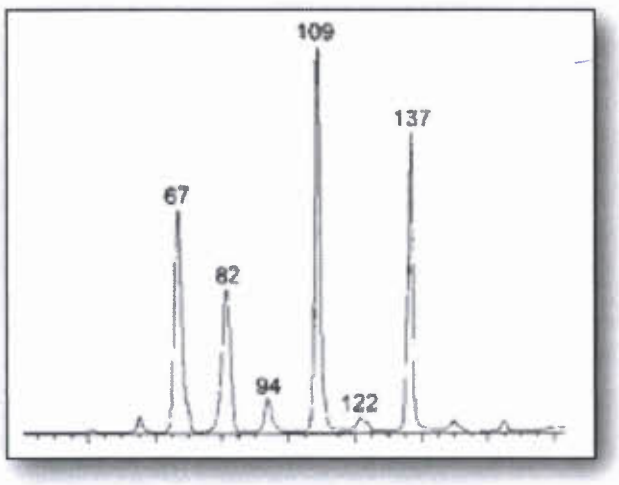
F) Comment le temps est-il représenté dans les systèmes répartis :
1. avec un diagramme de séquence
2. avec un graphe
3. avec un tableau d'historique d'exécution local

G) Needleman Wunsh est :
1. algorithme pour la génomique
2. algorithme de cryptographie
3. algorithme de gestion de données
4. algorithme pour les réseaux sociaux

H) L'algorithme utilisé par Google pour classer ses pages s'appelle
1. LBA Link Based Algorithm
2. pagerank
3. htmlrange
4. moneyrank

Exercice 2 (6 points)

L'analyse de substances chimiques constitue une étape primordiale de la recherche scientifique industrielle et fondamentale. La spectrométrie de masse indique des intensités différentes (ordonnées sur la figure ci-contre) pour une longueur d'ondes donnée (abscisses sur la figure ci-contre). Ainsi, chaque substance est caractérisée par des pics d'intensité donnés à des positions (longueur d'onde) données. On cherche à isoler automatiquement les pics issus d'une analyse par spectrométrie de masse.



Les informations sont stockées dans un tableau : `Intensite_tab` qui donne les valeurs d'intensités ordonnées par longueur d'onde.

1. Proposer un algorithme qui permet de saisir successivement plusieurs intensités du spectre qui seront saisies dans l'ordre des longueurs d'ondes et de les ranger au fur et à mesure dans un tableau `Intensite_tab`; L'utilisateur arrêtera la saisie en entrant une valeur négative. NB : les indices des tableaux commencent à 0.
2. Proposer un algorithme qui affiche respectivement (i) la valeur maximale, (ii) la valeur minimale et (iii) la valeur moyenne des intensités telles que stockées dans `Intensite_tab`.
3. Proposer un algorithme qui recherche les pics d'intensités et les stocke dans un nouveau tableau `Pic_tab`. La matrice sera ensuite affichée sous forme de courbe

4. Parmi les valeurs de pics stockées dans `Pic_tab` , on s'intéresse aux pics d'intensités supérieurs à une valeur seuil pour isoler uniquement les pics de grandes intensités. Pour cela, proposer une fonction qui prend en arguments: un tableau de pics d'intensité `Pic_tab` et une valeur numérique seuil (`seuil`), et qui retourne un tableau indiquant les pics d'intensité uniquement au dessus de la valeur seuil. Ex : un seuil de 70 isole les pics 109 et 137 dans la figure ci-dessus.

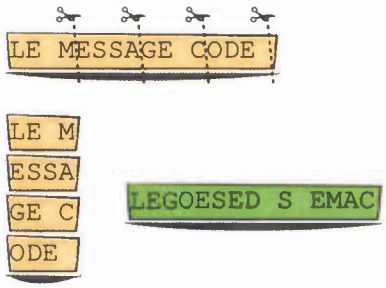
Exercice 3 (5 points)

Une chaîne de caractères représente un brin d'ADN si et seulement si elle est composée des caractères {'A', 'T', 'C', 'G'}, et seulement de ceux-ci. On veut vérifier qu'une chaîne de caractères, fournie par un utilisateur, peut représenter un brin d'ADN. On envisage deux méthodes algorithmiques. Méthode n°1 : Compter le nombre de caractères corrects et le comparer au nombre total de caractères de la chaîne. Méthode n°2 : Parcourir la chaîne et s'arrêter lorsque l'on peut décider si celle-ci ne contient que des caractères corrects ou au contraire contient au moins un caractère incorrect. On dispose des outils suivants : `longueur(chaine)` est une fonction qui prend une chaîne de caractères et renvoie le nombre de caractères qui la compose ; et `chaine[i]` représente le caractère situé à la position i. Chaque algorithme devra commencer par un commentaire précisant explicitement s'il considère que la position du premier caractère des chaînes est 0 ou 1.

1) Écrire un algorithme qui demande à l'utilisateur une chaîne censée représenter un brin d'ADN et qui vérifie si c'est le cas en suivant la méthode n°1. L'algorithme affichera l'une des deux phrases : « C'est un brin d'ADN" ou "Ce n'est pas un brin d'ADN, il y a x erreurs", où x est le nombre de caractères incorrects trouvés dans la chaîne.

2) Écrire une fonction qui demande à l'utilisateur une chaîne censée représenter un brin d'ADN et qui vérifie si c'est le cas en suivant la méthode n°2. La fonction retournera l'une des deux phrases : « C'est un brin d'ADN" ou "Ce n'est pas un brin d'ADN, première erreur en position y", où y est la position du premier caractère incorrect.

Exercice 4 (5 points)



Dans le cadre de la cryptographie, nous travaillerons sur le codage dit des «grecs», encore appelé Transcodage. De manière imagée, ce codage revient à écrire le message sur une bande de papier, à couper la bande de papier en morceaux de longueurs égales (cette longueur, exprimée en nombre de caractères est la clé), puis à mettre les bandes découpées les unes en dessous des autres pour lire le message codé (ici 'LEGOESED S EMAC'). Voici un exemple ci-contre:

D'un point de vue plus informatique, le découpage revient à lire les lettres du message clair dans un ordre déterminé en fonction de la clé pour construire le message codé. En gros pour une clé égale à 4 comme dans l'exemple, on lit une lettre sur 4 en faisant varier le point de départ (LEGO, puis ESED, puis...).

1) Ecrire, avec cette méthode, le mot codé obtenu à partir du message clair 'BONJOUR TOUT LE MONDE' avec une clé égale à 5.

2)Ecrire une fonction Encode qui prend deux paramètres, un message (chaîne de caractère) et une clé (un entier) et qui retourne le message codé (une chaîne de caractères) selon ce principe.

3) Expliquer comment la fonction Encode définie précédemment peut aussi être utilisée pour décoder le message (il s'agit principalement de donner le calcul de la clé de décodage à partir d'une clé d'encodage fixée).