

Informatique

Semestre 2

Examen — 1^{re} session

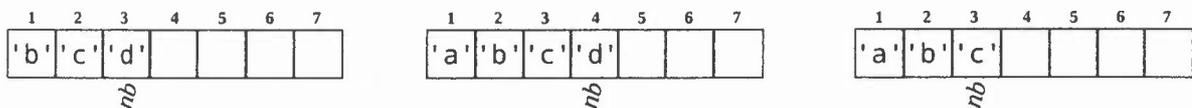
2013 Mercredi 9 janvier 2013, 14h00-15h30

Consignes : aucun document autorisé. Barème et durée des exercices donnés à titre indicatif.

Exercice 1 (File à double entrée ; 10 points, 45')

Le but de cet exercice est de définir une implémentation par tableau statique de la SDA *File à double entrée* de caractères, notée *FileDEC* dans la suite. Une file à double entrée est une file dans laquelle les données peuvent circuler dans les deux sens. Ainsi, un élément peut être enfilé avant le premier (opération *enfilerDevant*) ou après le dernier élément (opération *enfilerDerrière*). De même, on peut défiler le premier (opération *défilerDevant*) ou le dernier élément (opération *défilerDerrière*). On peut également accéder aussi bien au premier (opération *premier*) qu'au dernier élément (opération *dernier*). Enfin, on peut créer une *FileDEC* vide (opération *créer*) et tester si une *FileDEC* est vide (opération *estVide*).

L'implémentation par SDC *tableau statique* d'une *FileDEC* consiste à utiliser un tableau statique *elt* de N cases dont la case 1 représente toujours le premier élément et la case *nb* représente toujours le dernier élément.



La figure à gauche ci-dessus illustre une *FileDEC* contenant les caractères 'b', 'c' et 'd', dans cet ordre, dans un tableau statique de $N = 7$ cases. La figure centrale représente l'état de cette même *FileDEC* après qu'on ait *enfilerDevant* le caractère 'a'. La figure de droite représente toujours cette même *FileDEC* après qu'on l'ait *défilerDerrière*.

1. Donner et justifier la complexité temporelle en ordre de grandeur des opérations *enfilerDevant* et *défilerDerrière* telles que définies ci-dessous :

```

procédure enfilerDevant(FileDEC ref f,          procédure defilerDerriere(FileDEC ref f)
                    Caractere c)
// pre-condition : f.nb < N
// pre-condition : f.nb > 0
variable
    Entier i
debut
    pour i de 1 a f.nb faire
        f.elt[f.nb-i+2] ← f.elt[f.nb-i+1]
    finp
    f.elt[1] ← c
    f.nb ← f.nb+1
fin
    fin

```

2. Donner la définition du type *FileDEC* par tableau statique
3. Écrire la routine *créer* d'une *FileDEC*. Peut-elle servir à ré-initialiser une file ? Justifier.
4. Écrire les routines implémentant les opérations *estVide*, *enfilerDerrière*, *défilerDevant*, *premier* et *dernier* sur une *FileDEC*. Donner, en les justifiant, leurs pré-conditions.
5. Écrire la fonction *nbOcc* qui prend en paramètre une *FileDEC* f et un caractère c et retourne le nombre d'occurrences de c dans f . La complexité en temps de cet algorithme devra être en $\Theta(f.nb)$; le justifier.
6. Écrire le fichier entête *FileDECTabSta.hpp* donnant en C++ la définition du type *File* et la signature des routines agissant sur ce type dans un espace de noms *FileDECTabSta*.

Exercice 2 (Tri de chaînage par sélection ; 6 points, 30')

L'algorithme de tri par sélection d'un chaînage ch de réels consiste à considérer ce chaînage comme étant coupé en deux portions : entre les (pointeurs vers) maillons $ch.tete$ et $courant$ (exclus), ch est bien trié, alors qu'entre les (pointeurs vers) maillons $courant$ (inclus) et $ch.queue$, ch n'est pas encore trié. L'algorithme fait évoluer $courant$ de $ch.tete$ à $ch.queue$ en sélectionnant, à chaque itération, le plus petit élément dans la partie non-triée de ch , et en le permutant avec celui contenu dans le maillon (pointé par) $courant$. Sous l'hypothèse d'existence de deux routines annexes `minimum` et `échange`, l'algorithme peut donc s'écrire ainsi :

```
procedure triSelection (Chainage ref ch)
variables
    Pointeurs vers Maillon courant , min
debut
    courant ← ch.tete
    tant que courant ≠ ch.queue faire
        min ← minimum(courant)
        échange(courant , min)
        courant ← courant → succ
    fin tant que
fin
```

1. Écrire la fonction `minimum` qui prend en paramètre un pointeur vers maillon pm , parcourt tous les maillons chaînés depuis pm (inclus) et retourne un pointeur vers le maillon contenant le plus petit réel rencontré.
2. Écrire la procédure `échange` qui prend en paramètres deux pointeurs vers maillons et échange les réels qu'ils contiennent.
3. Calculer la complexité en temps de la procédure `triSelection` et indiquer son ordre de grandeur. Préciser la forme des données dans le meilleur et dans le pire des cas.

Exercice 3 (Questions ; 4 points, 15')

Répondre aux questions suivantes en quelques lignes ; les réponses doivent être bien justifiées :

1. Indiquer si $f(x) = 0.5x^2 - 3x + 2$ est en Ω , Θ ou O de $g(x) = x^2$.
2. Dans un chaînage simple, comment écrit-on l'accès à l'élément se trouvant deux maillons après un maillon m ? Quelles précautions faut-il prendre avant d'effectuer un tel accès ?

Examen X3I0030
Introduction aux systèmes d'information
Session 1 - Janvier 2013

N°anonymat: _____

parcours : _____

Remarques importantes

- La durée de l'épreuve est de : 1h30.
 - Le barème proposé est indicatif.
 - La clarté de votre présentation, l'écriture et l'orthographe seront prises en compte dans l'évaluation de votre copie.
 - Les documents papiers (dont les notes et polycopié de CM, TD et TP) sont interdits. Les livres, calculatrices et autres documents "électroniques" sont interdits.
 - Les schémas doivent être commentés, clairs et bien justifiés, faute de quoi ils seront comptés comme faux.
 - Les consignes énoncées en cours, en TD et en TP sont toujours valides. Le respect de ces consignes entrera en compte dans la notation.
-

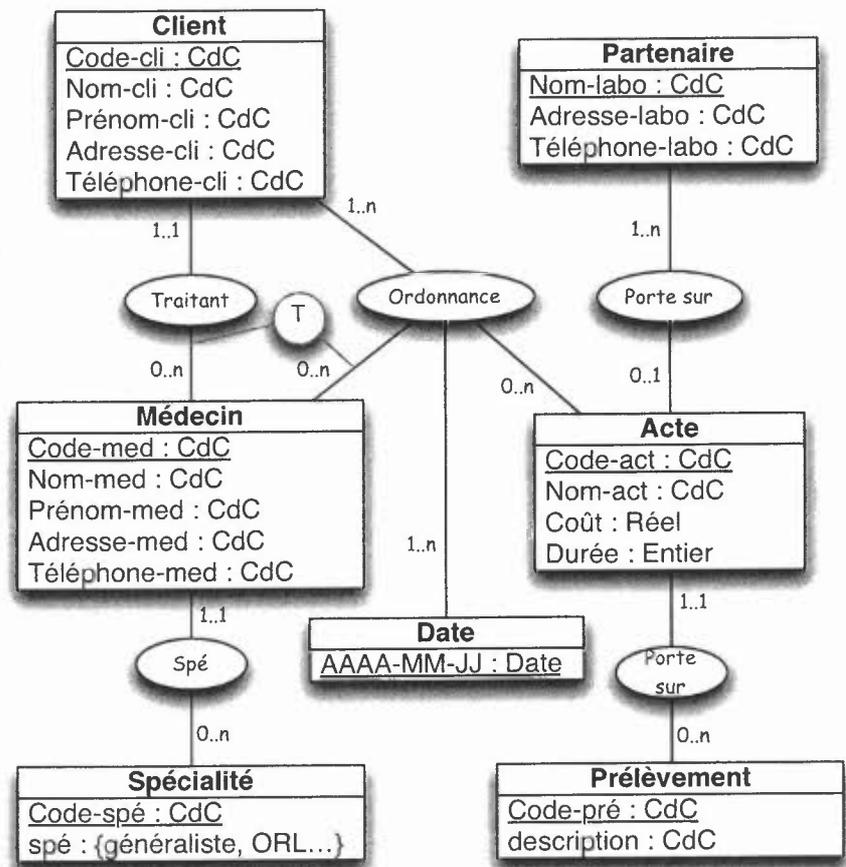
Note : / 20

Nombre de feuilles
complémentaires :

Commentaire :

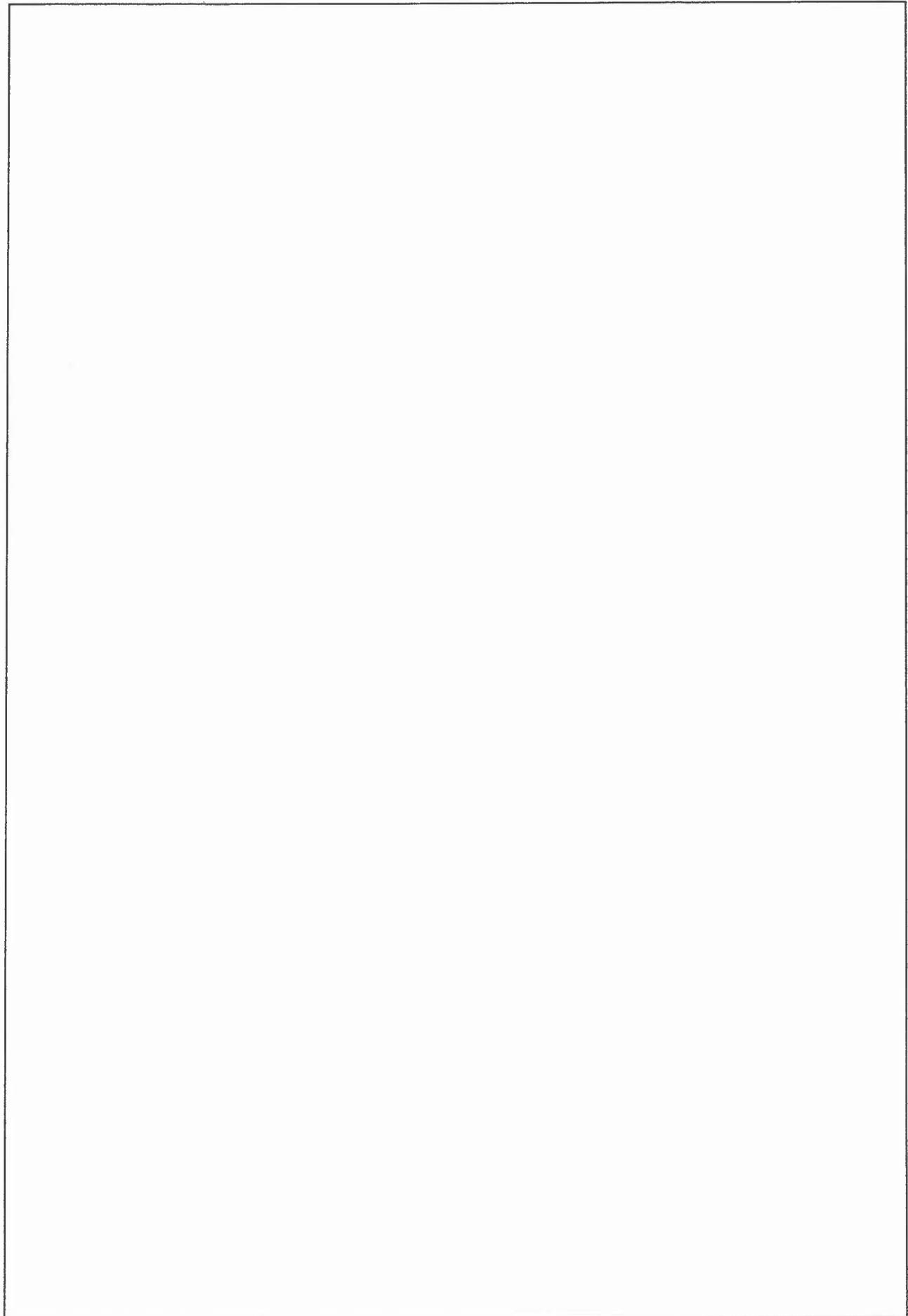
Exercice 1 - Laboratoire d'analyses médicales (7 points)

Un laboratoire d'analyses médicales cherche à mettre en place un système d'information pour mieux gérer les ordonnances et la diffusion des résultats. Les clients arrivent avec une ordonnance d'un médecin (un médecin traitant ou un spécialiste) comportant un certain nombre d'analyses à faire sur des prélèvements de sang, d'urine... Pour un client nouveau venu, son nom, son prénom, son numéro de téléphone, son adresse ainsi que son médecin traitant seront mémorisés. Par rapport à l'ordonnance, le médecin prescripteur sera ajouté, s'il n'est pas déjà présent. Un médecin, dont on a le nom, le prénom, le téléphone et l'adresse, peut être un généraliste ou un spécialiste comme un pneumologue, un ORL, etc. Une ordonnance est un ensemble de prescriptions concernant un client, un médecin et un acte à une date donnée. Il ne peut pas y avoir deux fois la même prescription sur une même ordonnance. Un acte est identifié par un code, une description, une durée standard et un coût (en euros). Certains actes sont sous-traités à des laboratoires partenaires (un laboratoire au maximum pour un acte). Les laboratoires partenaires ont un nom (identifiant) et une adresse. Un acte est effectué sur différents types de prélèvements possibles : sang, urine, tissu, selles, etc. Les résultats d'analyses seront envoyés au médecin traitant, au spécialiste éventuel ayant fait l'ordonnance et au client. Une première analyse donne le MCD ci-contre.



Question. En ne vous basant que sur le MCD, proposer un modèle relationnel permettant de modéliser les données concernées. Penser à préciser les contraintes d'intégrité référentielle.

Réponse :



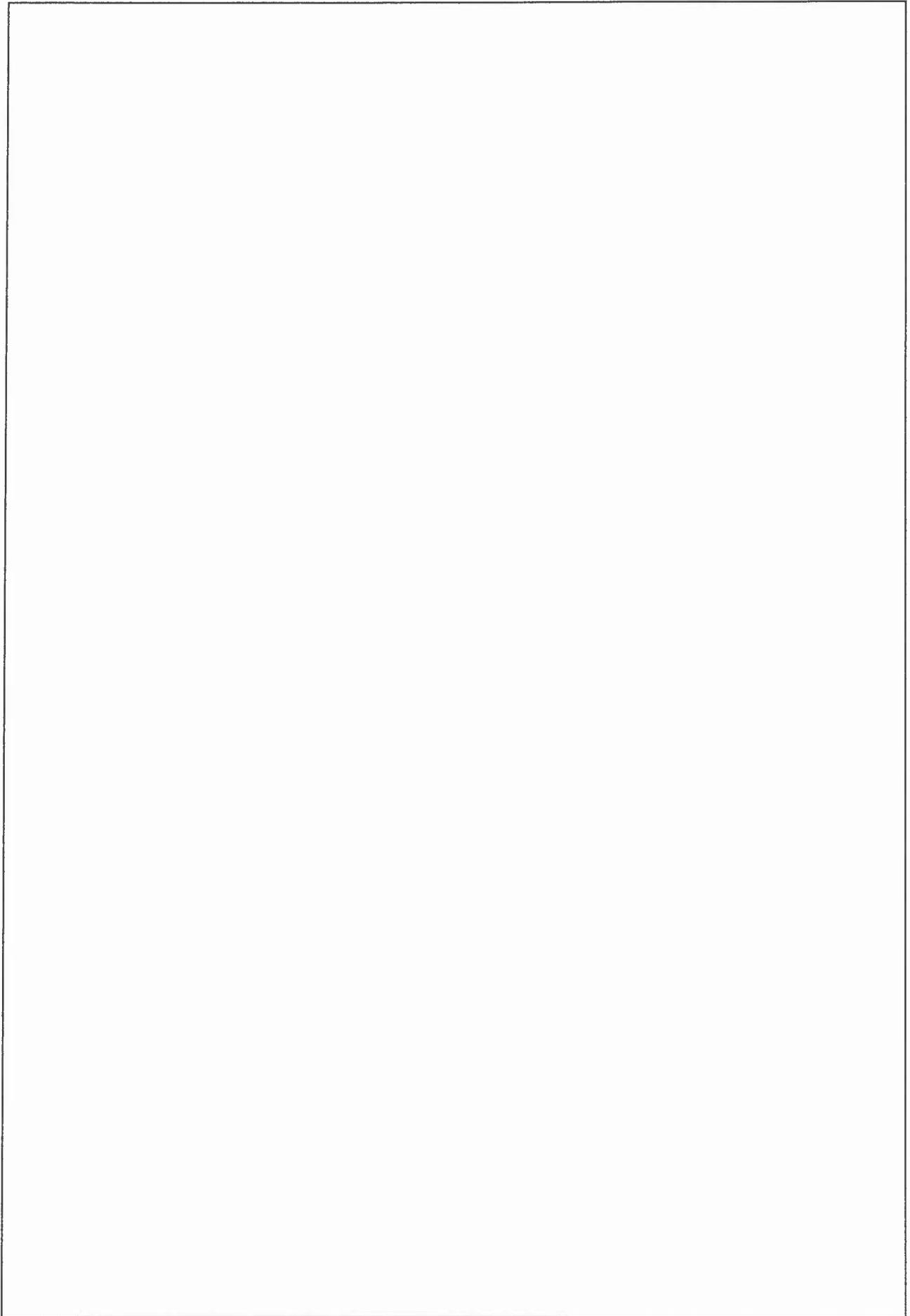
Exercice 2. ViteChezMoi.com (7 points)

Une jeune société de vente en ligne «ViteChezMoi.com» cherche à construire son système d'information commercial. Elle désire vendre des articles culturels et technologiques en ligne. Elle propose des articles matériels (livres, DVD, Blu-ray et CD) et des articles numériques (MP3, films numériques, livres numériques). Les premiers sont livrés à domicile dans les 48h après leur commande, alors que les seconds sont tout de suite téléchargeables en ligne. Pour ces derniers, une adresse de téléchargement sera proposée et le premier téléchargement sera mémorisé. Tout article numérique possède une version matérielle (ce n'est pas vrai dans l'autre sens). Il peut y avoir plusieurs formats numériques pour un même article matériel (par exemple, un livre peut être aussi en epub, en PDF et/ou en mobi). Les versions numériques sont accessibles par le client autant de fois qu'il le désire. Il peut même écouter la musique ou regarder le film qu'il a acheté en streaming. Chaque consultation en ligne est mémorisée (date, adresse IP) pour des calculs de charge du serveur. Chaque client peut donner un avis sur chacun les articles qu'il commande. Cet avis est composé d'une note sur 5 et d'un commentaire textuel. Nous supposons ici que tous les articles proposés font l'objet d'un même taux de TVA.

Question. En vous basant sur la description ci-dessus, proposer un modèle conceptuel des données qui seront manipulées. Mettre en évidence les problèmes rencontrés et les choix effectués. Attention aux points suivants :

- les fonctions ne sont pas représentées directement sur ce type de schéma ;
- bien mettre en évidence les différentes contraintes sur les données ;
- ne pas oublier le bon sens pour compléter avec des informations qui sont implicites.

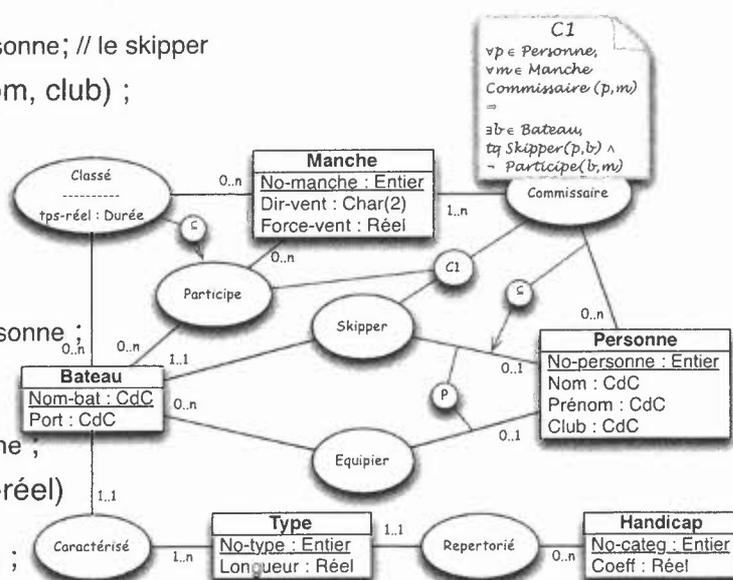
Réponse :



Exercice 3. Régates. (7 points)

On souhaite créer une base de données destinée à la gestion de régates¹. Les régates voient s'affronter des bateaux de différentes catégories. A chacune de ces catégories est affecté un handicap pour rendre compte des différences entre les bateaux. Un bateau est barré par un et un seul skipper. Il peut y avoir, selon la catégorie de navire, des équipiers. Un marin ne peut naviguer que sur un seul navire. Une régata est composée de manches. Les bateaux participant à une manche sont classés s'ils terminent la manche (ils peuvent aussi abandonner sur ennuis techniques par exemple). Leur temps est alors mémorisé. Durant les manches, des commissaires sont présents pour contrôler les différents concurrents (arbitrer l'arrivée et les virements aux bouées...). Les commissaires sont pris parmi les concurrents (skippers uniquement), ne concourant évidemment pas dans la manche qu'ils surveillent. L'analyse du projet amène la SSII chargée de la mise en œuvre à proposer le MCD et les tables suivantes :

- **Manche** (no-manche, dir-vent, force-vent) ;
- **Bateau** (nom-bat, port, #no-t, #no-personne)
avec Bateau.no-t -> Type.no-t
& Bateau.no-personne -> Personne.no-personne; // le skipper
- **Personne** (no-personne, nom, prénom, club) ;
- **Type** (no-t, long, #cat)
avec Type.cat -> Handicap.cat;
- **Handicap** (cat, coef) ;
- **Equipier** (#nom-bat, #no-personne)
avec Equipier.nom-bat -> Bateau.nom-bat
& Equipier.no-personne -> Personne.no-personne ;
- **Participe** (#nom-bat, #no-manche)
avec Participe.nom-bat -> Bateau.nom-bat
& Participe.no-manche -> Manche.no-manche ;
- **Classé** (#nom-bat, #no-manche, tps-réel)
avec Classé.nom-bat -> Bateau.nom-bat
& Classé.no-manche -> Manche.no-manche ;
- **Commissaire** (#no-manche, #no-personne)
avec Commissaire.no-manche -> Manche.no-manche
& Commissaire.no-personne -> Personne.no-personne .



Question. En utilisant SQL2 effectuer les recherches suivantes :

- Les listes des bateaux (nom du bateau) inscrits aux régates avec leur skipper (nom et prénom).

Réponse :

¹ Exercice inspiré d'un exercice de Jean-Bernard LEAUTE (IUT-Lannion).

- La liste des bateaux (nom) de type IMOCA (dont la longueur est de 60 pieds, soit 18,28 m), classés à au moins une manche, .

Réponse :

- Le nombre d'équipiers par bateau classés par port puis par nom du bateau.

Réponse :

- Pour chaque skipper (son nom et prénom), le nombre de fois où il a été commissaire.

Réponse :

Examen — première session

Indications : documents interdits sauf mémento algorithmique-C++; exercices indépendants pouvant être réalisés dans un ordre quelconque; barème donné à titre indicatif.

2h

Exercice 1 - *Simulation* (6 pts)

```
fonction pgcd (d entier x, d entier y) : entier
debut
1 si (y = 0) alors
2   retourner (x)
3 sinon
4   retourner pgcd(y, x MOD y)
5 fin si
fin

fonction saisie_positif() : entier
variables
  entier nb
debut
1 ecrire ("donner un entier positif")
2 lire (nb)
3 tant que (nb ≤ 0) alors
4   ecrire ("donner un entier positif")
5   lire (nb)
6 fin tant que
7 retourner (nb)
fin

// algorithme principal
variables
  entier a, b, resultat
debut
1 a <- saisie_positif()
2 b <- saisie_positif()
3 resultat <- pgcd(a, b)
4 ecrire ("le pgcd de ", a, " et ", b, " est ", resultat)
fin
```

Simuler cet algorithme selon la méthode tabulaire en supposant que l'utilisateur saisira successivement les trois nombres entiers : 0, 15, 6.

Exercice 2 - *Tempus fugit, hora volant* « *Le temps s'enfuit, les heures s'envolent.* » (7 pts)

Le temps est usuellement mesuré en heures, minutes et secondes.

1. Représenter un **instant (heures, minutes, secondes)** à l'aide d'une structure.
2. Il s'agit d'écrire un sous-algorithme permettant de convertir un instant en nombre de secondes sachant que une heure vaut 60 minutes et qu'une minute vaut 60 secondes.
 - a) Énoncer la signature du sous-algorithme.
 - b) Effectuer l'analyse de ce problème.
 - c) Énoncer le sous-algorithme.
3. Le but est d'écrire un sous-algorithme permettant de convertir un nombre de secondes en un instant.
 - a) Énoncer la signature du sous-algorithme.
 - b) Effectuer l'analyse de ce problème.
 - c) Énoncer le sous-algorithme.
4. Écrire un sous-algorithme permettant d'additionner deux instants.
 - a) Énoncer la signature du sous-algorithme.
 - b) Effectuer l'analyse de ce problème.
 - c) Énoncer le sous-algorithme.

Exercice 3 - *Croissant* (7 pts)

1. Il s'agit d'écrire un algorithme qui lit des entiers dans un fichier puis affiche ceux qui sont positifs en ordre inverse. À la fin, l'algorithme affiche le nombre d'entiers qui ont été lus dans le fichier, et le nombre d'entiers positifs. Le nom du fichier est choisi par l'utilisateur.

Exemple :

Le fichier comprend les nombres <-1, 12, -5, 2, 22, 23, 7>

Les nombres affichés sont : 7, 23, 22, 2, 12.

7 entiers ont été lus dont 5 positifs

- a) Effectuer l'analyse de ce problème en précisant bien les préconditions de votre solution et les structures de données.
- b) Énoncer l'algorithme.

2. Il s'agit d'écrire un algorithme qui lit des entiers dans un fichier puis affiche ceux qui sont supérieurs à la moyenne de ces nombres. À la fin, l'algorithme affiche le nombre d'entiers qui ont été lus dans le fichier, et la moyenne. Le nom du fichier est choisi par l'utilisateur.

Exemple :

Le fichier comprend les nombres <-1, 12, -5, 2, 22, 23, 7>

Les nombres affichés sont : 12, 22, 23.

7 entiers ont été lus. La moyenne est 8,57.

- a) Effectuer l'analyse de ce problème en précisant bien les préconditions de votre solution et les structures de données.
- b) Énoncer l'algorithme.



UNIVERSITÉ DE NANTES

U.F.R. des Sciences et des
Techniques

S.E.V.E. Bureau des Examens

Nom de l'U.E. : Informatique 4 Math-Éco

Code de l'U.E. : X410090

Date de l'examen : 10 janvier 2013

Durée : 2h00

Documents autorisés : aucun

Calculatrice autorisée oui non Type :

Année universitaire 2012-2013

Semestre 1 2

Session 1 2

A. COMPLEXITÉ [15 MIN - 3 POINTS]

Donner la forme des données au mieux et calculer le coût au pire (en fonction de t) d'un appel de la fonction ci-dessous. Détailler les calculs puis donner l'ordre de grandeur correspondant.

Fonction complement3(entier t, tableau d'entiers A)

Début

```
tot ← 0
```

```
pour ind de 1 à t faire // pour chaque case
```

```
tot ← tot + T[ind]
```

```
tant que tot mod 6 ≠ 0 faire // tant que tot n'est pas divisible par 6
```

```
tot ← tot + T[ind]
```

```
fin tant que
```

```
fin pour
```

Fin

B. EXÉCUTION [15 MIN - 4 POINTS]

Préciser pour chaque ligne du « main » l'affichage provoqué lors de l'exécution :

```
public class Chainage {
private class Maillon {String mot;int decal;Maillon suiv;}
Maillon courant;
Chainage(String base){
courant=new Maillon(); courant.decal=1; courant.mot=base; courant.suiv=courant;
}
public void ajoute(String txt, int delta){
Maillon ancien=courant.suiv;courant.suiv=new Maillon();
courant.suiv.mot=txt;courant.suiv.decal=delta;courant.suiv.suiv=ancien;
}
public String suivant(){ // attention : effet de bord
String res=courant.mot;int dep=courant.decal;
for (int i=0;i<dep;i++){courant=courant.suiv;}
return res;
}
public static void main(String[] args) {
Chainage texte=new Chainage("be"); System.out.println(texte.suivant());
texte.ajoute("bop", 1); System.out.println(texte.suivant());
texte.ajoute("a", 2); System.out.println(texte.suivant());
texte.ajoute("lula", 3); System.out.println(texte.suivant());
System.out.println(texte.suivant());
System.out.println(texte.suivant());
System.out.println(texte.suivant());
System.out.println(texte.suivant());
}
}
```

B. ÉCRITURE [50 MIN - 7 POINTS]

Un sac est une structure de données sans ordre mais avec répétition, c'est à dire que les éléments ne sont pas numérotés, et que le même élément peut être ajouté plusieurs fois. Écrivez deux implémentations en Java d'une structure de sac de chaînes de caractères disposant des méthodes :

```
SacC SacC() // fournit un sac vide (aucun élément) : constructeur

void mets(String txt) // ajoute un exemplaire de la chaîne txt au sac

int nbMax() // renvoie la quantité de la chaîne la plus présente
// renvoie 0 si et seulement si le sac est vide

String prend() // renvoie la chaîne présente le plus de fois dans le sac
// et l'enlève totalement du sac
```

Écrire deux implémentations d'une classe `SacC` comportant le constructeur et les 3 méthodes ci-dessus. On doit pouvoir utiliser cette classe comme dans l'exemple suivant. L'une des implémentations doit utiliser une liste chaînée, l'autre une `ArrayList.Piece`(reference, fabricant, materiau)

```
SacC bag=new SacC(); System.out.println(bag.nbMax()); // affiche 0
bag.mets("cha");System.out.println(bag.nbMax()); // affiche 1
bag.mets("cha");System.out.println(bag.nbMax()); // affiche 2
bag.mets("bada");bag.mets("bada");bag.mets("bada");
System.out.println(bag.nbMax()); // affiche 3
System.out.print(bag.prend()+" ");System.out.println(bag.nbMax()); // affiche bada 2
bag.mets("la");bag.mets("cha");bag.mets("la");System.out.println(bag.nbMax()); // affiche 3
System.out.print(bag.prend()+" ");System.out.println(bag.nbMax()); // affiche cha 2
```

D. COMPARAISONS [15 MIN - 2 POINTS]

Comparer dans un tableau le coût temporel des méthodes dans vos deux implémentations (abouties ou non ; on ne donnera que les ordres de grandeur des coûts). Discuter ensuite du coût spatial (place mémoire).

E. REQUÊTES [25 MIN - 4 POINTS]

On suppose disposer d'un accès à une base de données de pièces détachées automobiles. Les pièces détachées ont une référence et un fabricant. Elles sont parfois regroupées en entités (ensemble de pièces destinées à être utilisées ensemble). Deux pièces n'ayant pas le même fabricant ou pas la même référence peuvent être utilisées l'une à la place de l'autre (pièce alternative). Ces informations sont regroupées dans les tables suivantes :

`Piece`(reference, fabricant, materiau, nomPiece)

`Entite`(nomEntite, Vehicule, detaillable) detaillable est un booléen qui vaut faux si l'entite est toujours vendue entière

`FaitPartie`(#reference, #nomEntite)

`Remplace`(#reference1, #reference2)

Écrire les requêtes SQL permettant d'obtenir :

- 1) toutes les pièces du fabricant « Valeo » ; afficher le nom et la référence.
- 2) tous les fabricants de la pièce référence « EN357K »
- 3) toutes les pièces composant l'entité « Delco » de « Scenic »
- 4) toutes les pièces pouvant remplacer la pièce référence « EN357K »
- 5) tous les fabricants fournissant des pièces de l'entité « Delco » de « Scenic »

Remarque : Les temps et le barème précisés ne sont qu'indicatifs.

Semestre 2
Session 1

Exercice 4 – Signaux (3.5 pts)

Soit le programme suivant :

```
1 int main (void)
2 {
3     if ( fork() ) wait (NULL);
4
5     printf ("% d ] fini \ n " , getpid ());
6     return 0;
7 }
```

- 1) Expliquez le fonctionnement de ce programme.
- 2) En sachant qu'à la terminaison d'un fils le signal SIGCHLD est envoyé au père, réécrire ce programme sans utiliser l'appel système wait(). Par exemple, en utilisant l'appel système pause(), on pourra écrire une solution qui met le processus dans l'état endormi jusqu'à ce qu'un signal SIGCHLD soit reçu.

Exercice 5 – Ordonnement (3 pts)

Soient trois processus pour lesquels la date d'arrivée, la durée des calculs et la date à laquelle on attend *au plus tard* les résultats (date appelée *échéance*) sont données dans le tableau suivant :

Processus	Date d'arrivée	Durée de traitement	Echéance
P1	0	4	5
P2	0	5	12
P3	0	2	7

- 1) Dessiner le chronogramme d'exécution des processus selon la politique SJF non préemptif. Les échéances sont-elles respectées ?

Répondre sur la page 3 du sujet.

- 2) Indiquer les temps de rotation, les temps d'attente et les rendements individuels des processus sous SJF non préemptif.
- 3) En utilisant un algorithme d'ordonnement à priorités fixes, indiquez comment attribuer les priorités de façon à obtenir tous les résultats au plus tard à leur échéance (une valeur de priorité élevée impliquant une priorité élevée) ? Dessiner la séquence d'ordonnement produite par cette nouvelle politique.

Répondre sur la page 3 du sujet.

Indiquez ci-dessous vos réponses aux questions de l'exercice 5 :

SJF non préemptif

0	5				10				15				20				25			
P1																				
P2																				
P3																				

Ordonnancement à priorités

0	5				10				15				20				25			
P1																				
P2																				
P3																				



UNIVERSITÉ DE NANTES

U.F.R. des Sciences et des Techniques

S.E.V.E. Bureau des Examens

Année universitaire 2012-2013

Semestre 1 2

Session 1 2

Nom de l'U.E. :

Algorithmique et structures de données 2

Code de l'U.E. : S22I020

Date de l'examen :

15 mai 2013

Durée :

1 h 30

Documents autorisés :

non

Calculatrice autorisée

oui non

Type :

Remarques :

les temps et le barème précisés ne sont qu'indicatifs, les exercices ne sont pas classés par ordre de difficulté.

A.Complexité

[4 points – 15 min]

1.Forme

Préciser la forme des données au mieux et au pire pour un appel de la procédure process, la taille des données étant le paramètre p. Le tableau V est supposé contenir des entiers relatifs (négatifs, nuls ou positifs) mais p est un entier positif ou nul.

```

procédure process(tableau d'entiers V, entier p)
  locales ind,aux entiers
début
  ind ← 0
  tant que ind < p faire
    si V[ind] > V[p-ind-1]
      alors
        aux ← V[ind]
        V[ind] ← V[p-ind-1]
        V[p-ind-1] ← aux
      sinon
        ind ← ind + 1
    finsi
  fin tant que
fin

```

2.Coût

Détailler le coût au mieux de cette procédure puis donner (en le justifiant) un ordre de grandeur du coût moyen. On négligera le coût d'accès à un élément d'un tableau à partir de son indice.

B.Spécifications

[4 points – 15 min]

Ci-dessous est présentée une spécification de pile de réels :

Constructeurs :	Accesseurs :	Reconnaisseurs :
pV : → Pile	sommet : Pile → Réel	estVide : Pile → booléen
emp : Pile x Réel → Pile	dep : Pile → Pile	
Axiomes :		
① sommet(pV)=ε	③ dep(pV)=ε	⑤ estVide(pV)=VRAI
② sommet(emp(p, x))=x	④ dep(emp(p, x))=p	⑥ estVide(emp(p, x))=FAUX

3.Exécution

Appliquer les axiomes ci-dessus sur l'expression suivante jusqu'à n'avoir plus que des constructeurs. Détailler les calculs en précisant à chaque étape l'axiome utilisé.

```
dep(emp(dep(emp(emp(pV, sommet(dep(emp(emp(pV, 4), 3))))), 5)), 8))
```

4.Écriture

Spécifier (signature et tous les axiomes nécessaires) une fonction diff donnant la valeur obtenue en soustrayant à la première valeur empilée toute les autres (Remarque : le sommet est la valeur empilée en dernier). Par exemple après avoir empilé 5 puis 4 puis 7 dans la pile vide, la fonction diff fournira -6 :

diff(emp(emp(emp(pV, 5), 4), 7)) = -6 doit être un théorème démontrable avec ces axiomes.

5.Inventer

Voici une présentation d'une structure de données où chaque valeur est associée à un Acteur ; une telle structure peut servir à gérer l'accès à des ressources ou assurer la traçabilité d'une création, par exemple :

- Des ordinateurs ont accès à une zone de stockage temporaire, et déposent des fichiers ; l'acteur est alors le nom du programme.
- Des personnes travaillent ensemble sur un document texte partagé, l'acteur d'une modification est alors la personne qui l'a réalisée.
- Un moteur est assemblé à partir de pièces créées en différents endroits, l'acteur associé à chaque pièce est le nom de l'usine qui l'a fabriquée.

Proposer (en quelques lignes, sans écrire le code) trois façons d'implémenter une telle structure de données possédant les méthodes suivantes (présentées comme les méthodes d'une classe travaillant par modification). Rédiger un tableau de comparaison des coûts.

```
void ajoute(T v, Acteur lui)           // lui ajoute v
booléen dejaLa(T v)                   // VRAI ssi la valeur v est déjà présente
void supprime(T v, Acteur lui)        // supprime un exemplaire de v venant de lui
Acteur proprio(T v)                   // un acteur responsable de l'ajout de v
T ajoutDe(Acteur lui)                 // une valeur ajoutée par lui
Liste d'Acteurs clients()             // liste des propriétaires des valeurs présentes
```

6.Comparer

Préciser quelles seraient les conséquences dans vos 3 implémentations des décisions suivantes (séparément) ; en particulier écrire les PRE- et POST-conditions des méthodes dans chaque cas :

- une même valeur v ne peut être présente qu'une fois
- une même valeur v ne peut être présente qu'une fois pour un acteur donné (mais plusieurs fois avec des acteurs différents)
- un même couple (valeur, acteur) peut être présent plusieurs fois.

7.Programmer

Écrire en utilisant les méthodes proposées (donc hors de toute implémentation) l'algorithme d'une fonction comptant le nombre de valeurs ajoutées par un acteur donné.

D.Preuves**8.Hoare**

Effectuer des preuves détaillées (en précisant à chaque étape la règle utilisée) des triplets suivants :

① { } $x \leftarrow y$ and z ; $z \leftarrow x$ or y { $z=y$ } // x,y et z sont des variables booléennes

② { $a=b*q+r$ & $0 \leq r$ & $r < b$ } // ce sont toutes des variables entières
 si $r+1 < b$ alors $r \leftarrow r+1$ sinon $r \leftarrow 0$; $q \leftarrow q+1$ fin si
 { $a+1=b*q+r$ & $0 \leq r$ & $r < b$ }

③ { $x \geq 0$ & $d=1$ } // ce sont toutes des variables entières
 tant que $x > 0$ faire $d \leftarrow 1-d$; $x \leftarrow x-d$ fin tant que
 { $d=1$ } // Remarque : $(d=1 \ \& \ x \geq 0) \vee (d=0 \ \& \ x > 0)$ pourrait être un invariant de la boucle

9.Terminaison

Montrer que la fonction suivante termine. Remarque : jeter un œil sur $2*x+d$.

```
fonction decOuPas(x entier) : entier
début   d ← 1
        tant que x > 0 faire
            d ← 1 - d
            x ← x - d
        fin tant que
        retourner d
fin
```

10.C'est pas fini (exercice bonus : 1 point hors barème)

Montrer la terminaison de la fonction suivante.

```
fonction PasApas(x entier, b booléen) : entier
début si x > 0 alors si b alors retourner 1-PasApas(x,FAUX)
        sinon retourner PasApas(x-1,VRAI)
        fin si
        sinon retourner 1
    fin si
fin
```

Examen d'informatique

Exercice 1

Une image est constituée d'un seul segment.

Expliquer pourquoi en format vectoriel le segment apparaît après un zoom toujours avec la même qualité alors qu'en format bitmap la qualité est moins bonne.

Exercice 2

Écriture d'une page web sur la photosynthèse.

Description de la page à réaliser

- Les captures d'écran en annexe montrent la présentation générale de la page.
- La couleur du fond a pour code RVB «AAFFEE».
- Les 2 images sont dans les fichiers «Biosphère.jpg» et «Distribution.jpg». Elles sont affichées à l'écran dans la première ligne d'une table HTML avec dans les cellules en dessous les commentaires s'y rapportant.
- En dessous de cette table se trouve une autre table comme ci-dessous :

Écosystème	Biomasse (tC/ha)	productivité primaire (tC/ha/an)
Forêt tropicale	450	18,0
Forêt tempérée	300	12,5
Toundra	6	1,4

- La partie inférieure de la page permet à l'utilisateur d'effectuer des calculs :
À partir d'une surface donnée exprimée en ha et d'un type d'écosystème, le bouton «calculer» permet d'effectuer à partir du tableau précédent le calcul de la quantité de carbone de la surface détruite (en tonnes) ainsi que du temps de reconstruction nécessaire.
- Les liens [Formulaire de calcul](#) et [Revenir en haut](#) permettent le déplacement dans le document entre le haut et le bas de la page où se trouve la zone de calcul.

Questions

1. Écrire le code HTML de la page. Mettre les liens internes.

Vous n'êtes pas obligés d'écrire sur votre feuille la totalité des textes lorsqu'il sont trop long, mettre alors des points de suspension.

2. Nommer dans le html les objets nécessaires aux calculs. Écrire la fonction javascript effectuant ces calculs et affichant les résultats aux bons endroits dans la page. On supposera que l'utilisateur ne fait pas d'erreur de saisie.

3. Insérer l'appel de la fonction précédente au bon endroit et selon le bon événement.

4. Expliquer, sans écrire le code html et javascript, ce qu'il faudrait faire si l'on souhaite changer l'image «Biosphère.jpg» ainsi que son commentaire en dessous lorsque l'utilisateur clique sur l'image.

Captures d'écran

Lien vers le calcul du bas de page

Le haut de la page

Image «Distribution.jpg»

Image «Biosphère.jpg»

Commentaires des images

Menu déroulant pour le choix de l'écosystème

Photosynthèse

file:///Users/dominique/Desktop/Enseignements/L288-X480090/Cours:TD:TP%202021

OpenFrameworks Fac EDT Talence LMDE Piscines Geoportail La Forêt de Sélenia TV Sciences Diandaa

La biomasse sur la terre et les océans

Formulaire de calcul

Vocabulaire
Biomasse : masse de matière organique qui constitue un être vivant. elle est exprimée en masse carbone.
Productivité primaire : somme des biomasses fabriquées par l'ensemble des organisme chlorophylliens en un lieu donné pendant un an. Elle est exprimée en tonnes de carbone par hectare et par an ($tC \cdot ha^{-1} \cdot an^{-1}$)

Cette carte satellite a été capturée par le capteur SeaWiFS. Les couleurs donnent une indication de l'amplitude et de la distribution globale de la production primaire océanique et terrestre.

Distribution des principaux biomes terrestres.

Le bas de la page avec le formulaire de calcul

Calcul selon le type d'écosystème du temps de reconstruction d'une biomasse détruite :

Écosystème	Biomasse (tC/ha)	productivité primaire (tC/ha/an)
Forêt tropicale	450	18,0
Forêt tempérée	300	12,5
Toundra	6	1,4

Surface détruite en ha : 1 Choisir le type d'écosystème : Toundra

calculer

Tonnes de carbone de la surface détruite : 6 t Temps de reconstruction : 4 ans

Effacer

Revenir en haut

zones textes

Semestre 2
Session 2

Examen d'informatique

Exercice 1

Un fichier contient une image de 1100 x 1091 pixels. On imprime cette image sur papier avec une résolution à l'impression de 150 dpi. Quelle est la taille de l'image en cm une fois imprimée sur le papier ?

Exercice 2

Écriture d'une page web sur les alcènes et alcanes.

Description de la page à réaliser

- Les captures d'écran en annexe montrent la présentation générale de la page à l'ouverture (figure 1) et après un choix de l'utilisateur (figure 2).
- La couleur du fond a pour code RVB «#A9FFC9».
- Le menu déroulant permet de choisir entre «Alcènes» (choix 1) et «Alcanes (choix 2).
- Les boutons «2D» et «3D» permettent de choisir une visualisation de la molécule en deux ou en trois dimensions. À l'ouverture de la page, par défaut, le bouton 3D est sélectionné et le bouton 2D est non cliquable.
- Le bouton «Ajouter un carbone» ajoute à chaque clic un atome de carbone à la molécule courante en tenant compte du choix dans le menu. De même le bouton «Retirer un carbone» retire à chaque clic un atome de carbone. Pour les Alcènes le nombre de carbones doit rester supérieur ou égal à 2, pour les alcanes il doit rester supérieur ou égal à 1. Il n'y a pas de limitation supérieure. Ces boutons sont **désactivés** à l'ouverture de la page.
- Le bouton «Initialiser» remet la page dans son état initial.
- Le champ texte **formule brute** affiche automatiquement la formule de la molécule définie par l'utilisateur à l'aide de l'interface de la page. Ce champ n'est pas éditable.
- Seules les images des alcènes (resp. les alcanes) ayant un nombre de carbones entre 2 et 6 (resp. entre 1 et 6) sont disponibles. Au delà de 6 carbones l'image du fichier «Nondisponible.png» est affichée.
- L'image à l'ouverture de la page est dans le fichier «Alcanes-Alcènes.png»
- Les fichiers d'images pour les alcènes et les alcanes ont des noms construits de la manière suivante :
 - pour le 3D : «**nommolécule.png**». (il y en a donc 11 en tout construits de cette manière)
 - pour le 2D : «**nommolécule2D.png**». (il y en a également 11 en tout construits de cette manière)
- En bas de page, «Voir la nomenclature» est un lien qui ouvre le fichier pdf «Nomenclature.pdf»

Questions

1. Écrire le code HTML de la page y compris le lien et les noms des objets devant être utilisés en javascript.

Vous n'êtes pas obligés d'écrire sur votre feuille la totalité des textes lorsqu'ils sont trop longs (mettre alors des points de suspension).

2. Traduire en javascript l'algorithme de la fonction Affiche() donnée en annexe

3. Écrire les fonctions «AjouteCarbone()» et «RetireCarbone()» qui sont appelées lorsqu'on clique sur les boutons associés. Ces fonctions doivent afficher dans la page web les informations de la molécule.

4. Insérer les appels de fonction dans le html selon les bons événements

Captures d'écran

Figure 1
La page à l'ouverture

Menu déroulant proposant alcènes ou alcanes

Fichier « Alcanes-Alcènes.png »

Lien vers le calcul du bas de page

Boutons «Ajouter» et «Retirer»

Bouton «Initialiser»

Champ texte du nom

Les alcènes

Les alcènes sont des hydrocarbures insaturés, caractérisés par la présence d'au moins une double liaison covalente entre deux atomes de carbone. Ces liaisons sont toujours de type covalente normale parfaite. Les alcènes non cycliques n'ayant qu'une double liaison possèdent une formule brute de la forme C_nH_{2n} où n est un entier naturel supérieur ou égal à 2. L'alcène le plus simple est l'éthylène (nom usuel de l'éthène).

Les alcanes

Les alcanes sont des hydrocarbures saturés. Ils ne sont constitués que d'atomes de carbone (C) et d'hydrogène (H), liés entre eux par des liaisons simples, les atomes de carbone sont reliés à un nombre maximal d'atomes d'hydrogène — d'où le nom de « saturé ». Les alcanes non cycliques possèdent donc une formule brute de la forme C_nH_{2n+2} où n est un entier naturel non nul. À l'inverse les alcènes, les alcynes, et les composés aromatiques sont des hydrocarbures insaturés.

Choisir un type d'hydrocarbures : Choisir le type de visualisation : 2D 3D

Image de la molécule : Formule brute :

Nom de la molécule :

Ajouter un carbone Retirer un carbone Initialiser

Voir la nomenclature

Figure 2
La page avec les choix «Alcènes», «visualisation 2D» et 4 carbones

Choix Alcènes

Image 2D du butène

3 clics sur le bouton «Ajouter» → C4H8

Hydrocarbures saturés et insaturés

Les alcènes

Les alcènes sont des hydrocarbures insaturés, caractérisés par la présence d'au moins une double liaison covalente entre deux atomes de carbone. Ces liaisons sont toujours de type covalente normale parfaite. Les alcènes non cycliques n'ayant qu'une double liaison possèdent une formule brute de la forme C_nH_{2n} où n est un entier naturel supérieur ou égal à 2. L'alcène le plus simple est l'éthylène (nom usuel de l'éthène).

Les alcanes

Les alcanes sont des hydrocarbures saturés. Ils ne sont constitués que d'atomes de carbone (C) et d'hydrogène (H), liés entre eux par des liaisons simples, les atomes de carbone sont reliés à un nombre maximal d'atomes d'hydrogène — d'où le nom de « saturé ». Les alcanes non cycliques possèdent donc une formule brute de la forme C_nH_{2n+2} où n est un entier naturel non nul. À l'inverse les alcènes, les alcynes, et les composés aromatiques sont des hydrocarbures insaturés.

Alcènes Choisir le type de visualisation : 2D 3D

Image de la molécule : Formule brute : C4H8

Nom de la molécule : butène

Ajouter un carbone Retirer un carbone Initialiser

Voir la nomenclature

variables globales

NbCAlcenes ← 1; // nombre de carbone des alcènes (CH2 n'existe pas, on commence à 1)

NbCAlcenes ← 0; // nombre de carbone des alcanes

NomsAlcenes, NomsAlcenes : Vecteurs de 7 chaînes

NomsAlcenes[0] ← ""; NomsAlcenes[1] ← ""; NomsAlcenes[2] ← "éthène"; ...

NomsAlcenes[0] ← ""; NomsAlcenes[1] ← "méthane"; ...

Fonction Affiche()

lexique formule, dimension : chaîne, choix : entier

Début

choix ← saisie(menu)

Si Bouton2D=vrai **Alors** dimension ← "2D" **sinon** dimension ← "" **Finsi**

Si choix= 1 **Alors** // alcènes

Si (NbCAlcenes>=2 **Alors** // l'alcène existe

 formule ← "C"+NbCAlcenes+"H"+2*NbCAlcenes;

Si NbCAlcenes<=6 **Alors** // affichage de l'Image

 ImagePageWeb ← NomsAlcenes[NbCAlcenes]+dimension+".png"

 // affichage du nom

 NomPageWeb ← NomsAlcenes[NbCAlcenes]

Sinon { // ImagePageWeb non disponible

 ImagePageWeb ← "nondisponible.png"

 NomPageWeb ← ""

Finsi // fin si image disponible

 FormulePageWeb ← formule; // on affiche la formule

Finsi // fin si l'alcène existe

Sinon // alcanes

Si NbCAlcenes>1 **Alors** // l'alcane existe

Si (NbCAlcenes=1 **Alors** // un seul carbone

 formule ← "CH4"

Sinon

 formule ← "C"+NbCAlcenes+"H"+2*(NbCAlcenes+1)

FinSi

Si (NbCAlcenes<=6) **Alors** // affichage de l'Image

 ImagePageWeb ← NomsAlcenes[NbCAlcenes]+dimension+".png"

 // affichage du nom

 NomPageWeb ← NomsAlcenes[NbCAlcenes]

Sinon // ImagePageWeb non disponible

 ImagePageWeb ← "nondisponible.png"

 NomPageWeb ← ""

Finsi

 // affichage de la formule

 FormulePageWeb ← formule

Finsi // fin si l'alcane existe

FinSi // fin si alcènes

FIN // Fin fonction