Université M'hamed Bougara de Boumerdes
Faculté des Sciences
Département de Mathématiques
Deuxième année Licence Recherche Opérationnelle
Module : Structure de Données.

20 12/20 13

Semestre 03

Correction de l'examen Final de Structure de données [1]

Exercice $N°01 : 7points = 1 + 1 + 2 + 3$

Soient les deux fonctions mystère suivantes :

1. **Dérouler la fonction "*mystere1*" pour $n = 2$, pour $n = 3$.**
   mystere1(2)=5 ; mystere1(3)=14 ; (01 Point)
2. **Dérouler la fonction "*mystere2*" pour ($b = 2$ et $n = 2$), pour ($b = 2$ et $n = 3$).**
   mystere2(2,2)=7 ; mystere1(2,3)=15 ; (01 Point)
3. **En déduire ce que font ces fonctions.**
   $mystere1(n) = \sum_{i=0}^{n} i^2$; $mystere2(b, n) = \sum_{i=0}^{n} b^i$ (02 Point)
4. **Proposer deux fonctions itératives (sans la récursivité), qui font la même chose que les deux fonctions précédentes.**

Algorithme 1 Regardez_ces_fonctions

{*La première fonction (1,5 Point)*}

```
fonction mystere1(n :entier) :entier
var i,resultat1 :entier ;
resultat ⟵ 0;
pour i allant de 1 à n faire
   resultat1 ⟵ resultat + (i ∗ i);
fin pour
renvoyer(resultat1) ;
fin ;
```

{*La deuxième fonction (1,5 Point)*}

```
fonction mystere2(b :entier ; n :entier) :entier ;
var i,resultat2 :entier ;
resultat2 ⟵ 0;
pour i allant de 0 à n faire
   resultat2 ⟵ resultat2 + b ∧ i;
fin pour
renvoyer(resultat2) ;
fin ;
```

1. Veuillez me signaler d'éventuelles erreurs sur ma boite émail : madani.bezoui@gmail.com

Exercice $N°02 : 06\ points = 2 + 2 + 2$

Soient P une Pile d'entier, et F une File d'entier.

1. **Écrire une fonction "$copier(P : Pile) : File$" qui copie les éléments de la pile P dans la file F.**

Algorithme 2 copier (02 Point)

---

fonctions copier(P :Pile) :File
var P1 : Pile ;F :File ; $Tab[1, .., 100]$: *tableau d'entier*;
début
initialiser(F) ;{On l'a pas vue en cours ! j'en tiendrai compte !}
tantque $pile\_vide(P) = faux$ **faire**
  dépiler(P,val) ;
  empiler(P1,val) ;
fin tantque
tantque $pile\_vide(P1) = faux$ **faire**
  dépiler(P1,val) ;
  enfiler(F,val) ;
  empiler(P,val) ;
fin tantque
renvoyer(F) ;
fin ;

---

2. Écrire une fonction "$copier\_inverse(P : Pile) : File$" qui copie les éléments de la pile P dans la file F, mais cette fois dans l'ordre inverse.

Algorithme 3 copier_inverse (02 Point)

---

fonctions copier_inverse(P :Pile) :File
var P1 : Pile ;F :File ; $Tab[1, .., 100]$: *tableau d'entier*;
début
initialiser(F) ;{On l'a pas vue en cours ! j'en tiendrai compte !}
tantque $pile\_vide(P) = faux$ **faire**
  dépiler(P,val) ;
  enfiler(F,val) ;
  empiler(P1,val) ;
fin tantque
tantque $pile\_vide(P1) = faux$ **faire**
  dépiler(P1,val) ;
  empiler(P,val) ;
fin tantque
renvoyer(F) ;
fin ;

---

3. Écrire une fonction "*test(P :Pile) :bouléen*" qui renvoie "*vrai*" si la pile P est triée selon un ordre croissant et "*faux*" sinon.

Algorithme 4 test (02 Point)

```
fonctions test(P :Pile) :booléen
var rep :booléen ; P1 : Pile ; Tab[1, .., 100] : tableau d'entier;
début
rep ⟵ Vrai;
si pile_vide(P) alors
   écrire("Cette Pile est vide !") ;
sinon
   dépiler(P,x) ;
   empiler(P1,x) ;
   tantque ((pile_vide(P) = faux) et (rep)) faire
      dépiler(P,val) ;
      empiler(P1,val) ;
      si x < val alors
         rep ⟵ Faux;
      finsi
      x ⟵ val ;
   fin tantque
   tantque pile_vide(P1) = faux faire
      dépiler(P1,val) ;
      empiler(P,val) ;
   fin tantque
finsi
renvoyer(rep) ;
fin ;
```

Exercice $N°03 : 07$ $points = 1 + 2 + 2 + 2$

Algorithme exo2

```
type Liste=enregistrement
elt :entier ;
suivant : ↑ Liste ;
fin_enregistrement
var L :↑ Liste ;
```

Soit une liste linéaire chaînée d'entiers, supposée non vide, dont le pointeur vers la tête est L.

1. **Écrire une fonction "*somme(L) :entier*" qui renvoie la somme des éléments de la liste.**
2. **Écrire une fonction "*test(L) :booléen*" qui renvoie "vrai" si la liste est triée selon un ordre croissant et "faux" sinon.**
3. **Écrire une procédure "*permute(L)*" qui permute le premier et le dernier élément de la liste.**
4. **Écrire une procédure "*tri(L)*" qui trie la liste selon un ordre décroissant.**

Algorithme 5 Somme (01 Point)

```
fonction somme(L) :entier
var courant : ↑Liste ; S : entier;
début
courant ⟵ L ; S ⟵ 0 ;
tantque courant <> nil faire
   S ⟵ S + courant ↑ .elt
   courant ⟵ courant ↑ .suivant ;
fin tantque
renvoyer(S) ;
fin ;
```

Algorithme 6 test (02 Point)

```
fonction test(L) :booléen
var courant,prec : ↑Liste ; reponse : booléen ;
début
courant ⟵ L ; reponse ⟵ Vrai ;
tantque (courant ↑ .suivant <> nil et reponse) faire
   prec ⟵ courant
   courant ⟵ courant ↑ .suivant ;
   si prec ↑ .elt > courant ↑ .elt alors
      reponse ⟵ Faux
   finsi
fin tantque
renvoyer(reponse) ;
fin ;
```

Questions Bonus 02*points*

1. **Cet arbre est-t-il binaire ?**
   OUI, parce que chaque noeud (père) à au maximum deux fils.(01 Point)
2. **Quelle est son hauteur ?**
   Son hauteur est de 3. (0,5 Point)
3. **Donner deux fils et deux pères.**
   Deux pères : 100 et 200, (0,25 Point)
   Deux files : 10 et 20. (0,25 Point)

Mr. BEZOUI
Bon courage !

Algorithme 7 Permuter (02 Point)

```
procédure permute(L)
var courant : ↑Liste ; tmp : entier;
début
courant ⟵ L ;
tantque courant ↑ .suivant <> nil faire
   courant ⟵ courant ↑ .suivant ;
fin tantque
tmp ⟵ courant ↑ .elt ;
courant ↑ .elt ⟵ L ↑ .elt ;
L ↑ .elt ⟵ tmp ;
fin ;
```

Algorithme 8 tri (02 Point)

```
procédure tri(L)
var courant :↑Liste ;i,j,k :entier ; Tab[1, .., 100] : tableau d'entier;
début
courant ⟵ L ;i ⟵ 0
tantque (courant <> nil et reponse) faire
   i ⟵ i + 1 ; Tab[i] ⟵ courant ↑ .elt ;
   courant ⟵ courant ↑ .suivant ;
fin tantque
pour j allant de 1 à i faire
   pour k allant de j+1 à i faire
      si Tab[j] < Tab[k] alors
         tmp ⟵ Tab[j];
         Tab[i] ⟵ Tab[k];
         Tab[k] ⟵ tmp;
      finsi
   fin pour
fin pour
courant ⟵ L
pour j allant de 1 à i faire
   courant ↑ .elt ⟵ Tab[j];
fin pour
fin ;
```