

# contrôle ASD 2015 + la correction~unv~jijel ~ fighting !!

UNIVERSITE DE JIJEL

FACULTE DES SCIENCES EXACTES ET D'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

## Examen d'Algorithmique et Structures de Données Avancées

### Exercice 1 (4 pts):

Déterminez la complexité temporelle de l'algorithme suivant:

Début

$I \leftarrow 1$

$Nb \leftarrow 0$

Tant que  $(I \leq N)$  Faire

$J \leftarrow 1$

Tant que  $(J \leq N)$  Faire

Si  $(I = J)$  Alors

$Nb \leftarrow Nb + 1$

FSi

$J \leftarrow J + 1$

FTantque

$I \leftarrow I + 1$

FTantque

Fin

$$T(n) = 4n^2 + 7n + 3$$

### Exercice 2 (9 pts):

On suppose dans cet exercice que chaque expression arithmétique infixée est donnée sous forme d'une chaîne de caractères terminée par le caractère "#".

On suppose aussi que chaque expression est valide et que les nombres utilisés dans l'expression sont des entiers compris entre 1 et 9. De plus, on considère les opérateurs binaires suivants :  $\{+, -, *, /, ^\}$ .

1. Transformez l'expression infixée ci-dessous en une expression postfixée.

$$4 + 7 / (2 - (6 + 3)) * 9 ^ 5 + 8 \quad 147263+-195^++8+$$

2. Ecrivez une fonction qui teste si un caractère est un opérateur.
3. Ecrivez une fonction qui détermine la priorité d'un opérateur.
4. En utilisant une pile, écrivez une procédure qui permet de transformer une expression infixée valide en une expression postfixée. La fonction doit stocker la transformation dans une file.

### Exercice 3 (7 pts):

Donnez deux exemples différents d'arbres binaires qui ne sont pas parfaits.

Ecrivez une fonction itérative qui teste si un arbre binaire est parfait.

## Exercice 1: 14pts

$l \leftarrow 1$	2	0,25
$Nb \leftarrow 0$		
Tant que ( $l \leq N$ ) Faire	$N+1$	0,50
$J \leftarrow 1$	$N$	0,50
Tant que ( $J \leq N$ ) Faire	$N(N+1)$	0,50
Si ( $l = J$ ) Alors	$N^2$	0,50
$Nb \leftarrow Nb + 1$	$2N$	0,50
$J \leftarrow J + 1$	$2N^2$	0,50
$l \leftarrow l + 1$	$2N$	0,50
$T(N) = 4N^2 + 7N + 3$		0,25

## Exercice 2: 09pts

1. 4 7 2 6 3 + - / 9 5 ^ \* + 8 +

2. Fonction Operateur (Val C : caractère) : Booléen  
 Debut  
 Si ((C = '+') ou (C = '-') ou (C = '/') ou (C = '^') ou (C = '\*')) Alors  
 Operateur  $\leftarrow$  vrai  
 Sinon  
 Operateur  $\leftarrow$  faux  
 Fsi  
 Fin  
 Finfonction

3. Fonction PrioriteOperateur (Val C : caractère) : Entier  
 Debut  
 Si (C = '^') Alors  
 PrioriteOperateur  $\leftarrow$  3  
 Sinon  
 Si ((C = '/') ou (C = '\*')) Alors  
 PrioriteOperateur  $\leftarrow$  2  
 Sinon  
 Si ((C = '+') ou (C = '-')) Alors  
 PrioriteOperateur  $\leftarrow$  1  
 Sinon  
 PrioriteOperateur  $\leftarrow$  -1  
 Fsi  
 Fsi  
 Fsi  
 Fin



7pts  
Procedure **InfixToPostfix** (Val Expl : chaîne de caractères, Ref ExpP : File)

Variables

P : Pile

i : entier

Debut

CreerPileVide(p)

CreerFileVide(ExpP)

i ← 1

Tant que (Expl[i] ≠ '#') Faire

Si (Expl[i] = '(') Alors

Empiler(Expl[i], P)

Sinon

Si (Expl[i] = ')') Alors

Tantque (Sommptet(P) ≠ '(') Faire

Enfiler(Sommptet(P), ExpP)

Depiler(P)

FTantque

Depiler(P)

Sinon

Si ((Opérateur(Expl[i])) Alors

Tantque ((Non PileVide(p)) et (Opérateur(Sommptet(P)) et

(PrioritéOpérateur(Expl[i]) ≤ prioritéOpérateur(Sommptet(P))) Faire

Enfiler(Sommptet(P), ExpP)

Depiler(P)

FTantque

Empiler(Expl[i], P)

Sinon

Enfiler(Expl[i], ExpP)

Fsi

Fsi

Fsi

i ← i + 1

FTantque

Tantque (Non PileVide(p)) Faire

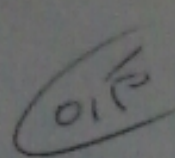
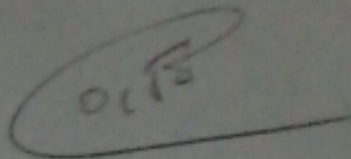
Enfiler(Sommptet(P), ExpP)

Depiler(P)

FTantque

Fin

FinProcedure

Exemple 1	Exemple 2
	
<p>2.</p> <p>Fonction <b>TestParfait</b> (Val A : ArbreBin) : Booléen</p> <p>Variables</p> <p>F : File      Parfait , FinParcours : Booléen</p> <p>Debut</p> <p>    parfait ← vrai</p> <p>    Si A ≠ nil    Alors</p> <p>        Enfiler(A,F)</p> <p>        CreerFileVide(F)</p> <p>        FinParcours ← faux</p> <p>        Tant que ((Non FileVide(F)) et (Non FinParcours)) Faire</p> <p>            A ← TeteFile(F)</p> <p>            Defiler(F)</p> <p>            Si (Degre(A) = 2 ) Alors</p> <p>                Enfiler(FilsGauche(A),F)</p> <p>                Enfiler(FilsDroit(A),F)</p> <p>            Sinon</p> <p>                Si (FilsGauche(A) ≠ Nil) Alors</p> <p>                    Enfiler(FilsGauche(A),F)</p> <p>                Sinon</p> <p>                    Parfait ← (Filsdroit(A) = Nil)</p> <p>                Fsi</p> <p>                FinParcours ← vrai</p> <p>            Fsi</p> <p>        FTantque</p> <p>        Tantque ((Non FileVide(F)) et (parfait)) Faire</p> <p>            Si (Feuille(TeteFile(F)) ) Alors</p> <p>                Defiler(P)</p> <p>            Sinon</p> <p>                Parfait ← faux</p> <p>            Fsi</p> <p>        FTantque</p> <p>    Fsi</p> <p>    TestParfait ← parfait</p> <p>Fin</p> <p>FinFonction</p> 