جامعة الشهيد حمه لخضر – الوادي –

كلية العلوم الدقيقة

قسم الإعلام الآلي

السنة الثالثة إعلام آلي

الأسئلة + الحلول النموذجية للامتحانات

السداسي الأول للموسم الجامعي 2017/2016

– الدورة العادية -

جامعة الشهيد حمه لخضر – الوادي –

كلية العلوم الدقيقة

قسم الإعلام الآلي

السنة الثالثة إعلام آلي

السداسي الخامس – الدورة العادية -

السنة الجامعية 2017/2016

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratiqueet Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جـــامـــعـــة الشهيد حمه لخضر – الـــوادي
Université Echahide Hamma Lakhdar – El-oued

Faculté des Sciences Exactes
Département d'informatique

كلية العلوم الدقيقة
قسم الإعلام الآلي

مقاييس امتحانات سنة ثالثة إعلام آلي – السداسي الخامس –

Pradgm Prgm..... الأستاذ : عباس مسعود
Pgm logique..... الأستاذ : لجدل إبراهيم
Compilation..... الأستاذ : لعويد عبد القادر
IHM..... الأستاذة : خلادي نجوى هدى
Système d'Exp, 2..... الأستاذ : برجوح شفيق
Pgm linéaire..... الأستاذ : كرثيو إسماعيل
Anglais..... الأستاذة : خلادي نجوى هدى
GL2..... الأستاذة : قطاس شروق

الموسم الجامعي **2017/2016**

الاسم اللقب الفوج

Université Ehahid Hamma Lakhdar ElOued

Faculté des Sciences Exactes 3 Informatique.

Département d'Informatique **EXAMENT de Paradigmes de Programmation** 02/02/2017

Exercice 01 (6pts) :

A) Classez les langages suivants suivant leurs paradigmes de programmation: **C++ Pascal Lisp HTML Java C OCaml XML C# Prolog FoCaLiZe PHP**

Impératif	Fonctionnel	Orienté Objet	Logique	Web

B) Complétez la table suivante par le **type fonctionnel** qui correspond :

`# fun a b -> a + b;;`	
`# (fun a b -> 3*a + b) 4 2;;`	
`# (fun a b -> a*b - 2) 4 ;;`	

`# let rec f l = match l with`
`| [] -> 0`
`| x::s -> (fun a b -> a + b) x (f s);;`	

`# [[(1, 2.5)]] ;;`	
`# (2, 5.5) :: [[(3, 9.5)]] ;;`	
`# [(1, 7.5)] @ [[(8, 6.3)]] ;;`	
`# let superC g a b = g a b ;;`	
`# superC (fun x y -> y + x) 5 8 ;;`	

Exercice 02 (5 pts)

Donnez la fonction OCaml **difference** (accompagnée de son **type**) qui calcule la différence ensembliste entre deux listes : **# difference L1 L2 :** tous les éléments de **L1** qui n'appartiennent pas à **L2.**

Exemple : # difference [1 ; 2 ; 3 ; 0] [3 ; 0 ; 5 ; 7] ;; retourne : int list = [1 ; 2].

Exercice 03 (9 pts)

Transformez le programme **Java** suivant en un programme fonctionnel **OCaml**.

JAVA	OCaml

```
class CompteCCP {


    private int Numero;
    private float Solde = 0.00;



    public CompteCCP(int N, float S)
    { Numero = N;
      Solde = S;}



    public void deposer(float  S)
    { Solde = Solde + S; }



    public void retirer (float  S)
    { Solde = Solde - S;}



    public void transferer
        (CompteCCP C ,  float S)
    {  retirer(S);  C.deposer(S);}



  }

  public static void ain(String[]args)
 {

CompteCCP C1 = new CompteCCP(1, 90.0) ;


CompteCCP C2 = new CompteCCP(2, 10.0) ;


C1.deposer(500.00);


C1.retirer(20.00);

C2.transferer(C1, 200.00);

  }
}
```

4 / 2

Université Ehahid Hamma Lakhdar ElOued
Faculté des Sciences Exactes
Département d'Informatique

3 Informatique
02/02/2017

COORIGE TYPE

EXAMENT de Paradigmes de Programmation

Exercice 01 (6pts) :

A) Classez les langages suivants suivant leurs paradigmes de programmation (table A) : **C++ Pascal Lisp HTML Java C OCaml XML C# Prolog FoCaLiZe PHP**

Table A

Impératif	Fonctionnel	Orienté Objet	Logique	Web
Pascal C	Lisp OCaml FoCaLiZe	C++ Java C#	Prolog	HTML XML PHP

B) Complétez la table suivante par le **type fonctionnel** qui correspond :

`# fun a b -> a + b ;;`	- : int -> int -> int = <fun>
`# (fun a b -> 3*a + b) 4 2;;`	- : int = 14
`# (fun a b -> a*b - 2) 4 ;;`	- : int -> int = <fun>

`# let rec f l = match l with`
`| [] -> 0`
`| x::s -> (fun a b -> a + b) x (f s);;`
	val f : int list -> int = <fun>

`# [[(1, 2.5)]] ;;`	- : (int * float) list list = [[(1, 2.5)]]
`# (2, 5.5) :: [[(3, 9.5)]] ;;`	Error
`# [(1, 7.5)] @ [[(8, 6.3)]] ;;`	Error
`# let superC g a b = g a b ;;`	val superC : ('a -> 'b -> 'c) -> 'a -> 'b -> 'c = <fun>
`# superC (fun x y -> y + x) 5 8 ;;`	- : int = 13

Exercice 02 (5 pts)

Donnez la fonction OCaml **difference** (accompagnée de son **type**) qui calcule la différence ensembliste entre deux listes : **# difference L1 L2** : tous les éléments de **L1** qui n'appartiennent pas à **L2.**

Exemple : # difference [1 ; 2 ; 3 ; 0] [3 ; 0 ; 5 ; 7] ;; retourne : int list = [1 ; 2].

```
# let rec appartient x l = match l with
  | [] -> false
  | y::r -> if (x=y) then true else appartient x r;;

val appartient : 'a -> 'a list -> bool = <fun>
```

```
#let rec difference l1 l2 = match l1 with
  | [] -> []
  | x::r -> if not(appartient x l2) then ::(difference r l2) else difference r l2 ;;

val difference : 'a list -> 'a list -> 'a list = <fun>
```

Exercice 03 (8 pts)

Transformez le programme Java suivante en un programme fonctionnel OCaml.

JAVA	OCaml
class CompteCCP {	Considérons un compte de type int*float Le type int répresente le numéro et le type float c'est le solde.
1 private int Numero; private float Solde = 0.00;	
2 public CompteCCP(int N, float S) { Numero = N; Solde = S;}	# let compteCCP (n:int) (s:float) = (a, b);; val compteCCP: int -> float -> int * float = <fun>
3 public void deposer(float S) { Solde = Solde + S; }	# let deposer ccp s = compteCCP (fst(ccp)) (snd(ccp)+.s);; val deposer : int * float -> float -> int * float = <fun>
4 public void retirer (float S) { Solde = Solde - S;}	# let retirer ccp s = compteCCP (fst(ccp)) (snd(ccp)-.s);; val retirer : int * float -> float -> int * float = <fun>
5 public void transferer (CompteCCP C , float S) { retirer(S); C.deposer(S);}	let transferer ccp1 ccp2 s = let x = (retirer ccp1 s) in deposer ccp2 s ;; val transferer : int * float -> int * float -> float -> int * float = <fun>

}
public static void ain(String[]args)
{	

6 CompteCCP C1 = new CompteCCP(1, 90.0) ;	# let c1 = compteCCP 1 90.0;;
7 CompteCCP C2 = new CompteCCP(2, 10.0) ;	# let c2 = compteCCP 2 10.0;;
8 C1.deposer(500.00);	# let c1 = deposer c1 500.00 ;;
9 C1.retirer(20.00);	# let c1 = retirer c1 20.00 ;;
C2.transferer(C1, 200.00); 10	# let c2 = transferer c1 c2 200.0 ;;
} }	

Université Hamma Lekhder d'El-Oued
Faculté des sciences exactes
Département d'informatique
3ème Licence informatique LMD
Module : PROgrammation LOGique

Session : Janvier 2017.
Durée : 1h30.
1 page.

Examen N° : 01

Aucun document autorisé.

Exercice 01: (07.5 points)

1. Écrire un programme prolog qui permet de tester l'appartenance d'un élément X à la liste L : appartient/2. (**2 points**)
2. Donnez l'arbre de résolution SLD de la requête : ?-appartient(X,[1,2,3]). (**4points**)
3. Déduire l'arbre de résolution de la requête : ?-appartient(2,[1,2]). (**1.5 points**)

Exercice 02 : (04.5 points)

Soit le programme Prolog suivant :

```
oiseau(pigeon).
oiseau(hirondelle).
carnivore(loup).
carnivore(lion).
animal(lion).
animal(X) :-oiseau(X).
manger(X,Y) :-carnivore(X),animal(Y), X\=Y.
```

1. Donner la solution de la requête : manger(lion,Y). (**1 points**)
2. Donner l'arbre complet de résolution SLD de but : manger(X,Y). (**2.5 points**)
3. Déduire l'arbre de résolution de la requête : manger(lion,Y). (**1 points**)

Exercice 03 : (08 points)

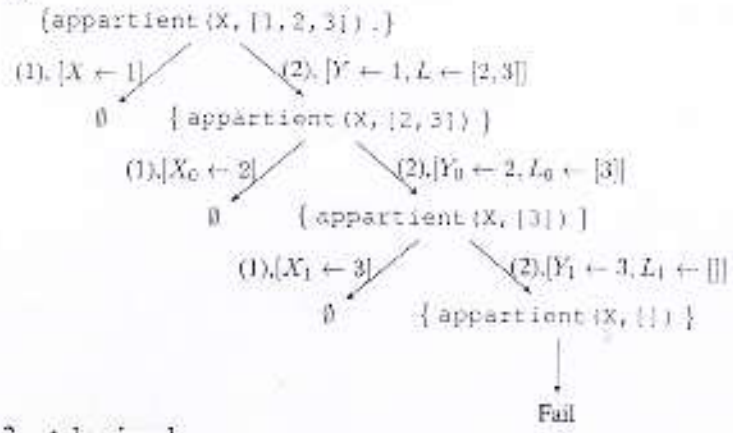
1. Écrire le prédicat aumoins(X,N,L) qui est vrai si l'entier X apparaît au moins N fois dans la liste d'entiers L. (**3 points**)
2. Écrire le prédicat supl(L,X) qui est vrai si X est plus grand ou égal à tous les éléments de la liste d'entiers L. (**2 points**)
3. Écrire le prédicat est_pair(X) qui est vrai si X est un entier pair (positif ou négatif). (**3 points**)

Bon courage

3/1

Correction d'Examen -Programmation Logique-

Exercice 01: (07.5 points)

1. Le programme est le suivant :
   (1) appartient(X,[X|_]).
   (2) appartient(X,[Y|L]):-appartient(X,L).
2. 

3. très simple.

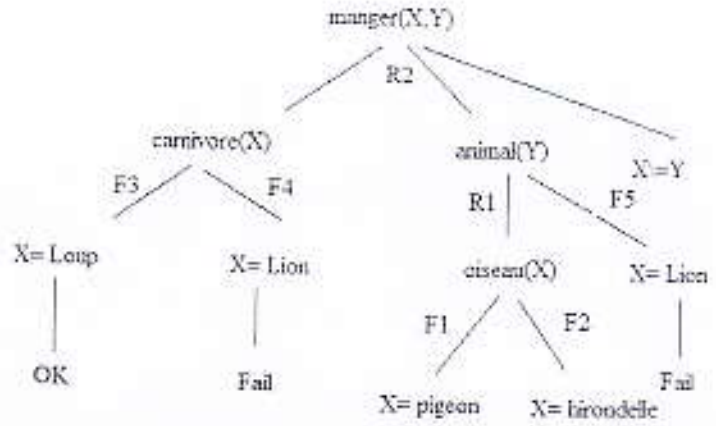
Exercice 02

1.
F1 oiseau(pigeon).
F2 oiseau(hirondelle).
F3 carnivore(loup).
F4 carnivore(lion).
F5 animal(lion).
R1 animal(X) :-oiseau(X).
R2 manger(X,Y) :-carnivore(X),animal(Y), X\=Y.

Par unification : manger(lion, Y) = manger(X,Y) / X= lion, Y = ?
On applique R2 : on doit vérifier carnivore(lion) et animal(Y), Y \= lion.
A partir F4 : carnivore(lion) est vrai il reste à vérifier animal(Y) tel que Y \= lion.
A partir R1 : on doit vérifier oiseau(X).
A partir F1 et F2 X= pigeon et X= hirondelle.

Donc :
Y = pigeon ;
Y = hirondelle.

2.



3.

Très simple, remplacez X par lion dans la réponse précédente. On annule la branche F5 car Y \= lion.

Exercice 03 : (08 points)

1.

```
aumoins(_,N,_):-N<=0.
aumoins(X,N,[X|L]):-M is N-1,aumoins(X,M,L).
aumoins(X,N,[Y|L]):-aumoins(X,N,L).
```

2.

```
supl([],X).
supl([X|L],Y):-Y>=X,supl(L,Y).
```

3.

```
est_pair(0).
est_pair(X):-X>0,Y is X-2, est_pair(Y).
est_pair(X):-X<0,Y is X+2, est_pair(Y).
```

La république algérienne

démochratique et populaire

Ministère d'enseignement supérieur et de la recherche scientifique

Université Echahid Hamma Lakhdar d'El-Oued
3ème année Liscence en informatique
Janvier 2017
Durée 01H30M: Examen de Compilation
Barém: 5 + 3 + 7 + 5 pts

Exercice 01 (Généralité):

Soit la grammaire G suivante:

R →(R) | R + R | R R | R * | a

a) Construire la dérivation la plus a gauche et la plus à droite pour la chaine (a+a)*a
b) Dessiner l'arbre de dérivation pour la chaine (a+a)*a
c) Décrivez le langage généré par la grammaire G
d) La grammaire G est elle ambiguë? La chaine (a+a)*a ambiguë?
e) Transformer la grammaire G en eliminant la récursivité a gauche en obtenant la grammaire G'
f) Dessinez l'arbre de dérivation pour la chaine (a+a)*a en utilisant la grammaire G'

Exercice 02 (Analyse LL(1)):

Soit la grammaire G suivante tel que les symboles terminaux sont {id, ", +}:

S → id | " T "
T → S V
V → ε | + S V

1. Calculer les ensembles First et Follow de chaque non terminal
2. Donner la table LL(1) de G
3. A partir de Pile: #S, Mot: " id + id " montrez l'évolution de la pile et le mot à lire en utilisant la table LL(1)

Exercice 03 (Analyse SLR):

Soit la grammaire G suivante:

(1) E →E + T
(2) E →T
(3) T →T F
(4) T →F
(5) F →F *
(6) F →a
(7) F →b

a) Calculer First et Follow pour chaque non terminal
b) Donner la table SLR de G
c) Montrez le déroulement pour analyser le mot w = a+ab*#

Exercice 04 (Question de cours) QMC (+/-) 0.5:

1. Le compilateur est plus lent que l'interpréteur
2. L'interpéteur génère un code exécutable
3. La phase de synthèse et de production précède la phase d'analyse
4. Éliminer les caracteres superflus doit être efectue durant l'analyse syntaxique
5. L'analyseur syntaxique produit un arbre syntaxique
6. L'analyseur lexical constitue la premiére étape d'un compilateur
7. Toute grammaire non récursive à gauche et une grammaire LL(1)
8. C'est possible de trouver ε dans un ensemble de Follow
9. Avant de commencer à éliminer la récursivité gauche, il faut que la grammaire soit proper
10. Une grammaire ambigüe produise deux actions de décalage dans la même case dans la table SLR

5/2

La république algérienne

démocratique et populaire

Ministère d'enseignement supérieur et de la recherche scientifique

Université Echahid Hamma Lakhdar d'El-Oued
3ème année Liscence en informatique
Janvier 2017
Durée 01H30M: Examen de Compilation
Barème: **5 + 3 + 7 + 5** pts

Exercice 01 (Généralité):

Soit la grammaire G suivante:

R →(R) | R + R | R R | R * | a

a) Construire la dérivation la plus à gauche et la plus à droite pour la chaine (a+a)*a
b) Dessiner l'arbre de dérivation pour la chaine (a+a)*a
c) Décrivez le langage généré par la grammaire G
d) La grammaire G est elle ambiguë? La chaine (a+a)*a ambiguë?
e) Transformer la grammaire G en éliminant la recursivité à gauche, en obtenant la grammaire G'
f) Dessinez l'arbre de dérivation pour la chaine (a+a)*a en utilisant la grammaire G'

Exercice 02 (Analyse LL(1)):

Soit la grammaire G suivante tel que les symboles terminaux sont {id, *, +}:

S → id | * T
T → S V
V → ε | + S V

1. Calculer les ensembles First et Follow de chaque non terminal
2. Donner la table LL(1) de G
3. À partir de Pile: #S, Mot: * id + id montrez l'évolution de la pile et le mot à lire en utilisant la table LL(1)

Exercice 03 (Analyse SLR):

Soit la grammaire G suivante:

(1) E →E + T
(2) E →T
(3) T →T F
(4) T →F
(5) F →F *
(6) F →a
(7) F →b

a) Calculer First et Follow pour chaque non terminal
b) Donner la table SLR de G
c) Montrez le déroulement pour analyser le mot w = a+ab*#

Exercice 04 (Question de cours) QMC (+/-) 0.5:

1. Le compilateur est plus lent que l'interpréteur
2. L'interpéteur génère un code exécutable
3. La phase de synthèse et de production précède la phase d'analyse
4. Éliminer les caractères superflux doit être effectué durant l'analyse syntaxique
5. L'analyseur syntaxique produit un arbre syntaxique
6. L'analyseur lexical constitue la première étape d'un compilateur
7. Toute grammaire non récursive à gauche est une grammaire LL(1)
8. C'est possible de trouver ε dans un ensemble de Follow
9. Avant de commencer à éliminer la récursivité gauche, il faut que la grammaire soit proper
10. Une grammaire ambiguë produise deux actions de décalage dans la même case dans la table SLR

5/2

Exo 2

$R \xrightarrow{1} (R)$
$R \xrightarrow{2} R+R$
$R \xrightarrow{3} RR$
$R \xrightarrow{4} R*$
$R \xrightarrow{5} a$

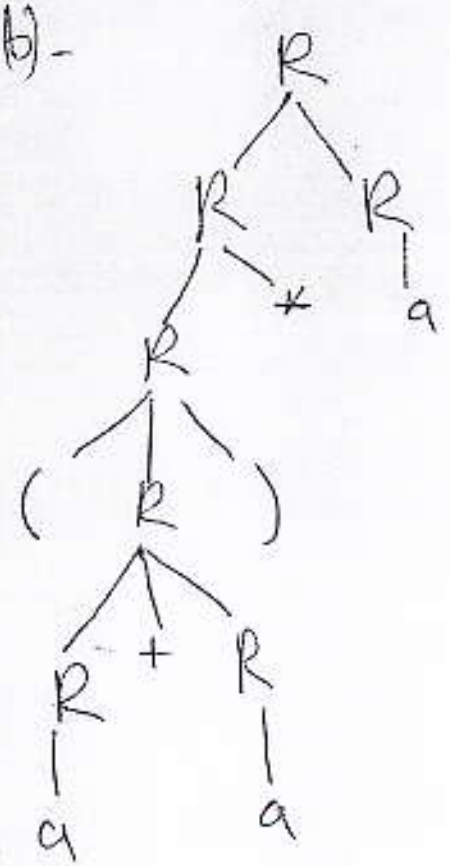
a) 1- DLG :

$R \xrightarrow{3} RR \xrightarrow{4} R*R \xrightarrow{1} (R+R)*R \xrightarrow{1} (a+R)*R \xrightarrow{1} (a+a)*R \xrightarrow{1} (a+a)*a$

2- DLD

$R \xrightarrow{3} RR \xrightarrow{5} Ra \xrightarrow{4} R*a \xrightarrow{1} (R)*a \xrightarrow{2} (R+R)*a \xrightarrow{5} (R+a)*a \xrightarrow{5} (a+a)*a$.

b) -



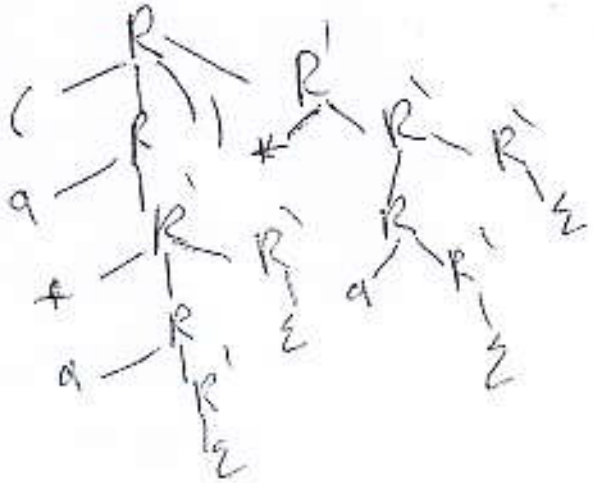
c- Est un langage génère toutes les operations d'addition et multiplication imprigiées

d) Nom, et la chaine (a+a)*a aussi non ambiguë.

e)

$R \to (R)\,R' \mid a\,R'$
$R' \to +R\,R' \mid R\,R' \mid *R' \mid \varepsilon$

f)



Exo 2

First(S) = { id, " } First(T) = { id, " }

First(V) = { ε, + }

Follow(S) = { #, + / " }

Follow(T) = { " }, Follow(V) = { " }

	id	"	+	#
S	S→id	S→"T"		
T	T→SV	T→SV		
V		V→ε	V→+SV	

Pile	Mot	Règle
#S	"id+id"	S→"T"
#"T"	"id+id"	T→SV
#"VS	id+id"	S→id
#"Vid	id+id"	V→+SV
#"VS+	+id"	S→id
#"Vid	id"	V→ε
#"	"	

Exo 3

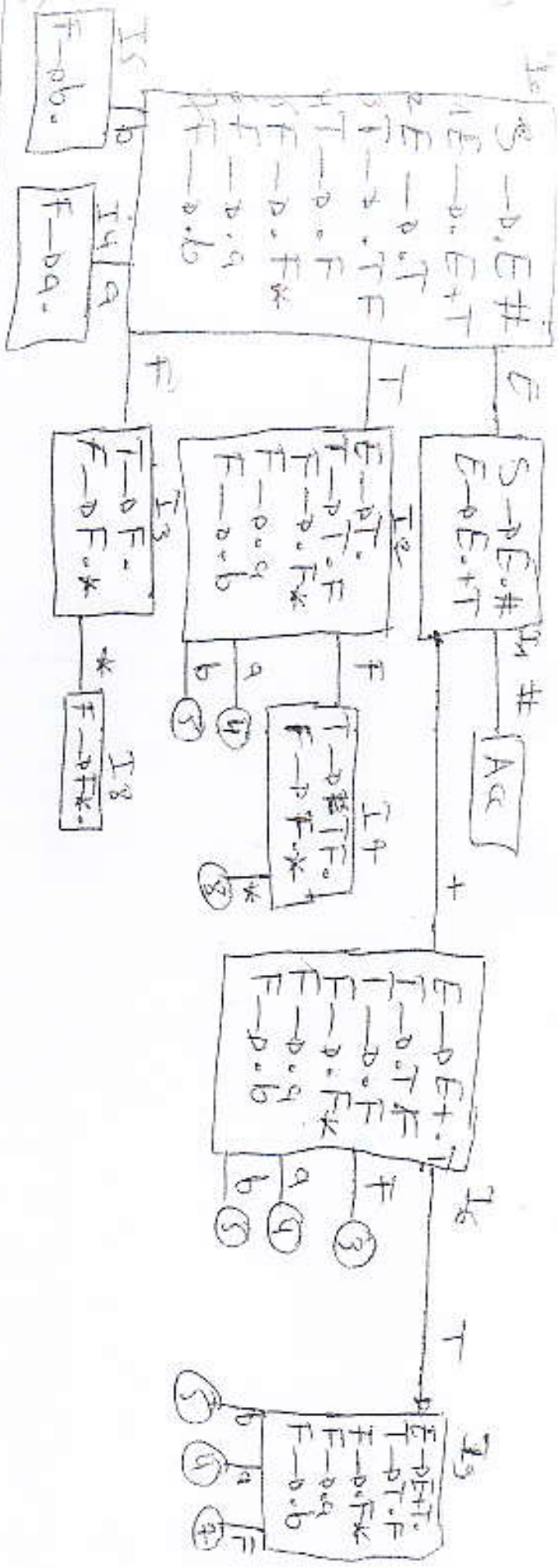
a) First(E) = {a, b} = First(T)

= First(F)

Follow(E) = {#, +}

Follow(T) = {#, +, a, b}

b) Follow(F) = {#, +, a, b, *}



	a	b	*	+	#	E	T	F
I_0	d4	d5				g01	g02	g03
I_1				d6	Acc			
I_2	d4	d5		r2	r2			g07
I_3	r4	r4	d8	r4	r4			
I_4	r6	r6	r6	r6	r6			
I_5	r7	r7	r7	r7	r7			
I_6	d4	d5					g09	g03
I_7	r3	r3	d8	r3	r3			
I_8	r5	r5	r5	r5	r5			
I_9	d4	d5		r1	r1			g07

c) Voir TD

Exo4

1- g
2- f
3- f
4- f
5- g
6- g
7- f
8- f
9- g
10- f

Université Hamma Lakhdhar Eloued — Corrigé Type IHM

Durée : 1h 30mn — 3eme A Informatique

Exercice 1 « 12 points »

1. $T = 0,1 \log_2 \frac{2D}{L}$ « **1 pt** »ça sera T = 0.8 s ; « **1 pt** »

2. - Reformuler l'information, -Ajouter du sens (raconter une histoire), - Imagination visuelle (visual thinking), - Organiser (chunking : Créer un mnème), - Faire des liens avec des connaissances existantes (catégories). « **3 pts** »

3. Le but des règles ergonomique pour avoir une compatibilité entre la machine et la psychologie de l'être humain (Homme).

4. Elle vise la compréhension fondamentale des interactions entre les humains et les autres composantes d'un système, et l'application de méthodes, de théories et de données pour améliorer le bien-être des personnes et la performance globale des systèmes. « **2 pts** »
5. raccourcis claviers s'effectue par Clavier « **0.5 pt** »; Touches d'accès s'effectue par souris ou clavier ; « **0.5 pt** »
6. rapidité d'accès ; facilité l'exécution ; réduction de temps d'exécution ; « **2 pts** »
7. les différents modèles en IHM : Modèle GOMS, Modèle Keystroke , Modèle Nielsen « **2 pts** »

Exercice 2

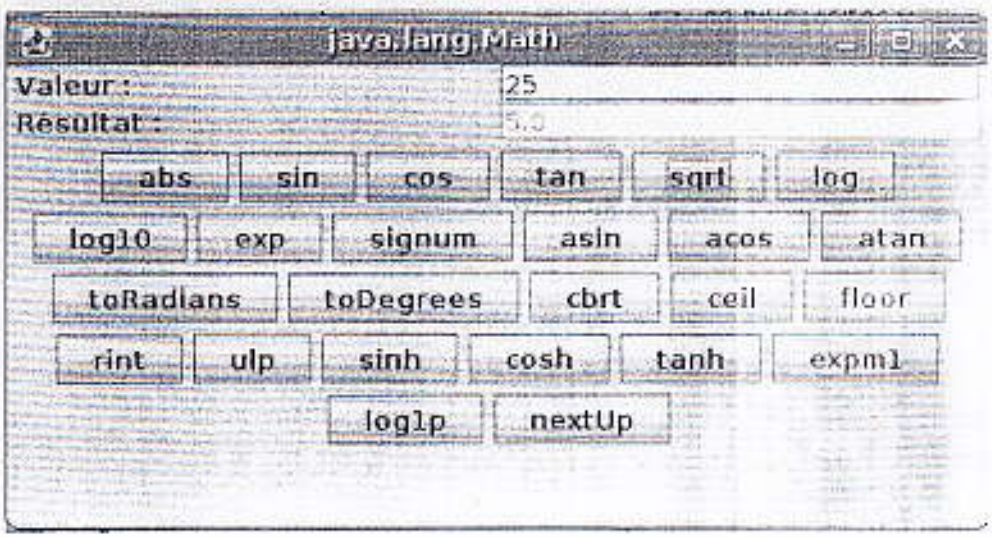
a. « **3 points** »
Barre d'outils inexistante.
Le titre de la fenêtre doit être changé.
Des raccourcis clavier doivent être ajoutés pour les commandes importantes (copier, coller, couper...)
Dans le menu < Format >, Pour Choisir Majuscule ou Minuscule on doit utiliser des < Radio Button >.
La commande < Police " doit être déplacé vers le menu << Format >. « **1 pt** »

« 2 pts »

b. « 3points »

- Prototype « 1 pt »
- Un prototype permet d'évaluer le fonctionnement du logiciel, il donne une idée sur l'importance du développement du logiciel et la spécification précise et définitive. Il sert de contrôle de qualité dans le but de détecter les erreurs. Il n'y a pas nécessairement une évolution continue de prototype vers le produit final par raffinement / extension. « 2 pts »

c. « 2points »



Université El-Oued
Faculté de sciences exactes
Département Informatique

3ème Année Informatique
Module : SE II
Durée : 1h30 - Le : 08/01/2017

Contrôle Semestriel

*La clarté et la propreté de la copie rendue sera notée sur **1pt**.*

Exercice 1 — 04 pts

1- Qu'est ce qu'une section critique ? Quelles conditions faut il vérifier afin d'assurer une bonne coopération entre les processus concernés par cette section critique?

2-Est-ce qu'il peut y avoir plusieurs sections critiques dans le code d'un processus ? Si oui ça dépend de quoi ?

Exercice 2 — 06.50 pts

1) Les trois *threads* A, B et C suivantes sont exécutées de manière concurrente ; elles utilisent pour leur synchronisation deux sémaphores S1 et S2.

La valeur initiale de S1 est 1, celle de S2 est 0 et celle de x est 0.

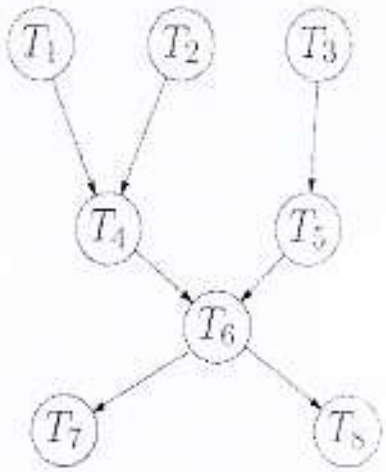
a. Quelles sont les différentes valeurs possibles de la variable x à la fin de l'exécution complète des trois *threads* ?

Thread A {	Thread B {	Thread C {
P(s2);	P(s1);	P(s1);
x = x * 2;	x = x * x;	x = x + 3;
V(s1); }	V(s1); }	V(s2); }

b. Qu'est-ce que cela changerait si la valeur initiale de S1 était 2 au lieu de 1 ?

2) Soit le graphe de précédence suivant :

Ecrire le code qui permet de faire une synchronisation entre ces tâches en utilisant des sémaphores (nombre minimum).



Exercice 3 — 04 pts

On reprend le problème du point de rendez-vous vu en TD. Proposez une solution à ce problème en utilisant les moniteurs.

1/1

Exercice 4 — 04.50 pts

On considère 4 processus P1, P2, P3, P4 et 3 types de ressources R1, R2, R3. Les tableaux ci-dessous expriment les besoins en ressources des processus pour leur exécution complète ainsi que les disponibilités totales de ressources des trois types :

besoins

	R1	R2	R3
P1	3	2	2
P2	6	1	3
P3	3	1	4
P4	4	2	2

disponibilités

R1	R2	R3
9	3	6

A un instant donné, le système est dans l'un des états suivants. Indiquez si ces états sont sains ou non sains et dans l'affirmative, indiquer la suite d'états permettant d'exécuter complètement les processus.

état 1

	R1	R2	R3
P1	1	0	0
P2	6	1	2
P3	2	1	1
P4	0	0	2

état 2

	R1	R2	R3
P1	1	0	0
P2	5	1	1
P3	2	1	1
P4	0	0	2

état 3

	R1	R2	R3
P1	2	0	1
P2	5	1	1
P3	2	1	1
P4	0	0	2

Bonne chance

Corrigé-Type

Exercice 1 — 04 pts

1) Une section critique (SC) est un ensemble de suites d'instructions qui peuvent produire des résultats imprévisibles lorsqu'elles sont exécutées simultanément par des processus différents. (2.5 pts)

Conditions : Exclusion Mutuelle, Progression, Attente bornée, Aucune hypothèse...

2) Oui, il est possible d'avoir plusieurs SC dans le code, cela dépend du nombre et de l'utilisation des ressources partagées. (1.5 pts)

Exercice 2 — 06.50 pts

1)

a- Le fait que S1 soit initialisée à 1 permet deux successions d'exécutions différentes pour les différents threads :

- Thread B, puis Thread C et enfin Thread A, ce qui donne x = 6.
-Thread C, puis Thread A et enfin Thread B, ce qui donne x = 36. (02 pts)

b- Si on initialise S1 à 2, alors les threads B & C peuvent s'exécuter simultanément et la valeur de x deviendra imprévisible et sûrement incohérente. (01 pts)

2) *5 Sémaphores sont nécessaires.* (03.50 pts)

```
Var s1 ;s2 ;s3 ;s4 ;s5 ;s6 ;s7 ;s8 : semaphore ;
   T1 : {// Travail ; V(s4); }
   T2 : {// Travail ;V(s4); }
   T3 : {// Travail ; V(s5); }
   T4 : {P(s4);P(s4);// Travail; V(s6); }
   T5 : {P(s5); // Travail ; V(s6); }
   T6 : { P(s6);P(s6); // Travail ; V(s7); V(s8); }
   T7 :  { P(s7); // Travail }
   T8 :  {P(s8); // Travail }
{T1//T2//T3//T4//T5//T6//T7//T8}
```

Exercice 3 — 04 pts

Structure du moniteur :

Type Rendez_Vous = Monitor Const N = 10 Var I : integer ; X : Condition
Procedure Entry Arrivee Begin I:= I + 1; If I = N Then X.Signal End

Procedure Entry Franchir Begin X.Wait; End
Begin /* Initialisation */ I := 0 End.

Structure des processus : RV étant une instance du type de moniteur Rendez_Vous.

Processus Pi	Processus Maitre
Début	Début
...	...
RV_Arrivee	RV_Franchir
...	...
Fin	Fin

Exercice 4 — 04.50 pts

- L'état 1 est un état sain. On peut, en effet opérer de la manière suivante :
  On donne 1 unité R3 à P2 ce qui lui permet de terminer son exécution. On récupère alors les ressources de P2.
  On donne 2 unités R1, 2 unités R2, 2 unités R3 à P1 ce qui lui permet de terminer son exécution. On récupère alors les ressources de P1.
  On donne 1 unité R1 et 3 unités R3 à P3, ce qui lui permet de terminer son exécution. On récupère alors ses ressources.
  On donne 4 unités R1 et 2 unités R2 à P4 ce qui termine tous les traitements.
- L'état 2 est également un état sain : on peut donner à P2 les ressources qui lui manquent et ensuite les récupérer, ce qui permettra, comme dans le cas de l'état 1 d'aller au terme des exécutions.
- L'état 3 est un état non sain. En effet, on ne peut donner à un processus les ressources qui lui permettraient de se terminer. On a donc un interblocage.

الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

جامعة حمه لخضر بالوادي
كلية العلوم الدقيقة
قسم الإعلام الآلي

السنة الدراسية 2017/2016
المدة: ساعة ونصف
السنة الثالثة

امتحان السداسي الأول في مقياس: Programmation linéaire

Exercice 01: (11 points)

Soit le problème linéaire suivant :

$$\begin{aligned} &MaxZ = 3x_1 - 5x_2 \\ &4x_1 + 5x_2 \geq 3 \\ &6x_1 - 6x_2 = 7 \\ &x_1 + 8x_2 \leq 20 \\ &x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

1) Résoudre ce programme linéaire par la méthode de simplexe à deux phases.
2) Justifiez votre réponse par une représentation graphique du problème.

Exercice 02: (09 points)

Un paysan possède trois firmes de pomme de terre (Hassi Khalifa, Tagzout et Rabah) dont la production est de 450, 350, 350 tonnes respectivement. Il veut vendre son produit dans trois marchés (Guemar, El Oued et Debila) dont les demandes sont 280, 500 et 370 tonnes respectivement. Les coûts de transport entre les différentes firmes et les marchés sont représentés dans le tableau suivant :

	Guemar	El Oued	Debila
Hassi Khalifa	40	40	10
Taghzout	30	20	30
Rabah	50	20	40

Le paysan décide de suivre la formulation de vente suivante:

a) La production de Hassi Khalifa est partagée comme suit : 280 tonnes vers le marché Guemar et le reste vers le marché de Debila.
b) Toute la production de Taghzout est vendu au marché d'El Oued.
c) La production de Rabah est partagée comme suit : 150 tonnes vers le marché d'El Oued et le reste vers le marché de Debila.

Question:

1) Est-ce que le problème admet un plan de transport optimal ?
2) Est-ce que le paysan fait la bonne formulation? sinon utilise la formulation du paysan comme base, pour obtenir le plan de transport optimale.

--بالتوفيق--

H/1

KO1:

Programmation linéaire

1) Résolution du PL par simplexe à deux phases

$\text{Max } 3x_1 - 2x_2$

$4x_1 + 2x_2 \geq 3$
$6x_1 - 6x_2 = 7$
$x_1 + 8x_2 \leq 20$
$x_1, x_2 \geq 0$

⟹ forme standard

$\text{Max } 3x_1 - 2x_2$
$4x_1 + 2x_2 - e_1 = 3$
$6x_1 - 6x_2 = 7$
$x_1 + 8x_2 + e_2 = 20$
$x_1, x_2, e_1, e_2 \geq 0$

On cherche une solution dual réalisable

$\text{Max } W = -a_1 - a_2$
$4x_1 + 2x_2 - e_1 + a_1 = 3$
$6x_1 + 8x_2 + a_2 = 7$ (1)
$x_1 + 8x_2 + e_2 = 20$
$x_1, x_2, e_1, e_2, a_1, a_2 \geq 0$
$-a_1 = 4x_1 + 2x_2 - e_1 - 3$
$-a_2 = 6x_1 - 6x_2 - 7$
$\Rightarrow W = -a - a_2 = 10x_1 - x_2 - e_1 - 10$ (1)
$W - 10x_1 + x_2 + e_1 = -10$

	x_1	x_2	e_1	e_2	a_1	a_2	b
a_1	(4)	2	-1	0	1	0	3
a_2	6	-6	0	0	0	1	7
e_2	1	8	0	1	0	0	20
D_j	-10	1	1	0	0	0	-10

(1)

	x_1	x_2	e_1	e_2	a_1	a_2	b
x_1	1	1/2	-1/4	0	1/4	0	3/4
a_2	0	(23/2)	3/2	0	-3/2	1	5/2
e_2	0	23/4	1/4	1	-1/4	0	77/4
D_j	0	27/2	-3/2	0	5/2	0	-5/2

1,5

	x_1	x_2	e_1	e_2	a_1	a_2	b
x_1	1	0	0	0	1/23	1/23	7/6
e_1	0	1	2/3	0	-1/23	2/23	5/3
e_2	0	0	-6	1	1/23	-11/23	113/6
D_j	0	0	0	0	1	1	0

1,5

Tous $D_j \geq 0$ et $W = 0$ on a une solution de base réalisable pour Z (0,5)

VB : x_1, e_1, e_2
VHB : x_2

$x_1 = x_2 = \frac{7}{6} \Rightarrow x_1 = x_2 + \frac{7}{6}$

$Z = 3x_1 - 2x_2$
$Z = 3\left(x_2 + \frac{7}{6}\right) - 2x_2 = 3x_2 + \frac{7}{2} - 2x_2$
$\Rightarrow Z + 2x_2 = \frac{7}{2}$ (0,5)

	x_1	x_2	e_1	e_2	b
x_1	1	-1	0	0	7/6
e_1	0	-9	1	0	5/3
e_2	0	9	0	1	113/6
D_j	0	2	0	0	7/2

(1)

Tous $D_j \geq 0$
donc
$Z^* = \frac{7}{2}$ (0,5)
$x_1 = 7/6$
$x_2 = 0$

2). Justification graphique :

(0,5) $D_1 : 4x_1 + 5x_2 = 3$: $(\frac{3}{4}, 0)$, $(0, \frac{3}{5})$

(0,5) $D_2 : 6x_1 - 6x_2 = 1$: $(\frac{7}{6}, 0)$, $(0, -\frac{1}{6})$

(0,5) $D_3 : x_1 + 8x_2 = 20$: $(0, \frac{5}{2})$, $(4, 2)$

$D^\circ : 3x_1 - 5x_2 = 0$, $(0,0)$, $(5,3)$

L'intersection entre A_1 et A_2

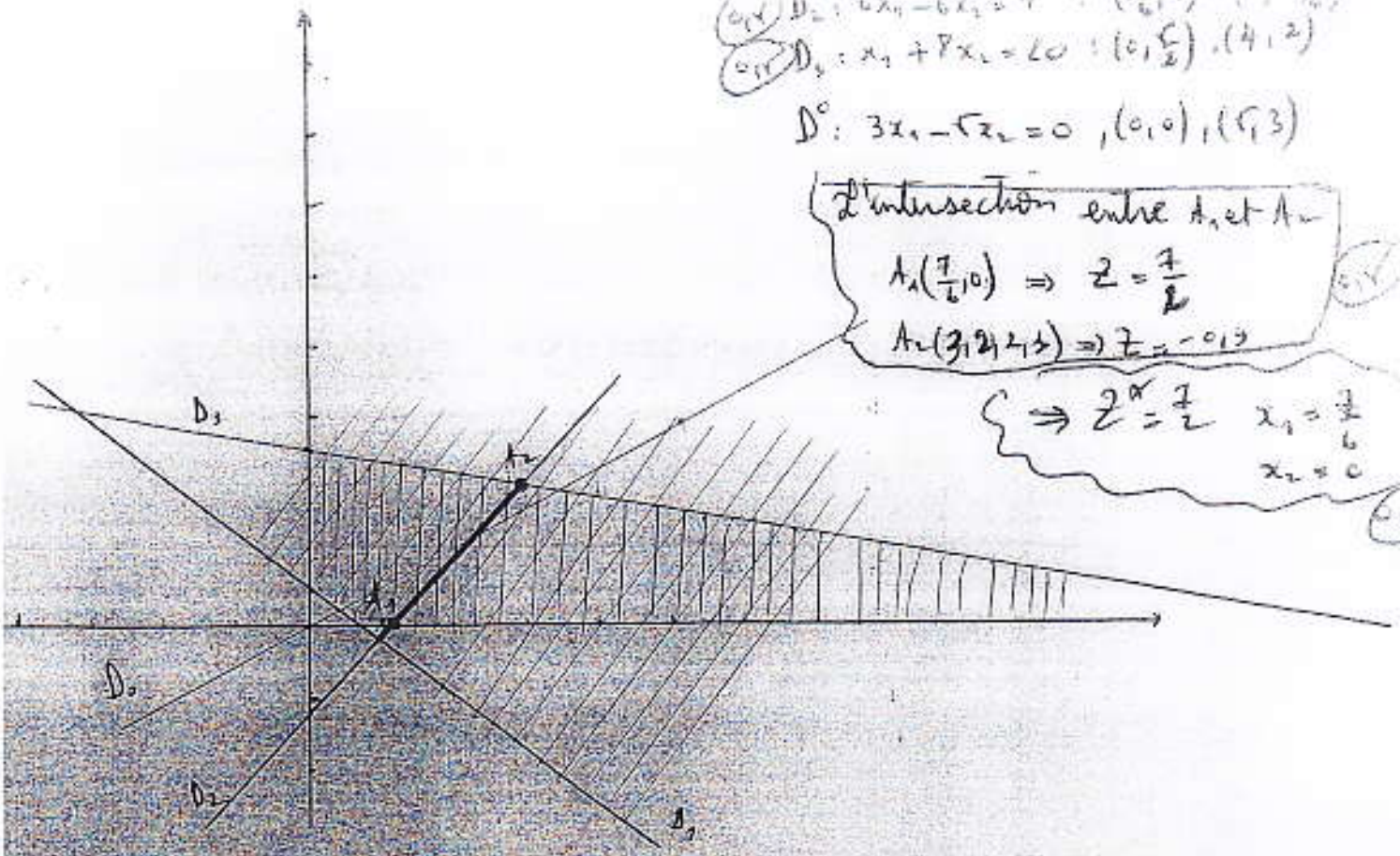
$A_1(\frac{7}{6}, 0) \Rightarrow Z = \frac{7}{2}$

$A_2(3,2 ; 2,2) \Rightarrow Z = -0,3$

(0,5)

$\Rightarrow Z^* = \frac{7}{2}$ $x_1 = \frac{7}{6}$ $x_2 = 0$

(0,5)



Ex 02 :

1) Que le problème admet un plan de transport optimal car la condition de balance est vérifié $\sum a_i = \sum b_j = 1150$ (2)

2) Pour vérifié si le paysan fait la bonne formulation il faut vérifié que :

a) La formulation est une base ; (1)

Si à toute la production des fermes sont acheminés et toute les demandes sont sont satisfaites alors la formulation est une base

b) Il faut que cette base soit l'optimale.

On utilise l'algorithme des potentiel est on teste si toutes le Δ_{ij} sont négatif ou égale a zéro

	GE	EO	DE	a_i	u_i
HK	40 − 280	40 ~~280~~ 40	10 + 170	450	0
TA	10 + 60	20 − 350	30 10	350	30
RA	50 20	20 + 150	40 − 200	350	30
b_j	280	500	370		1150
v_j	40	−10	10		

D'après cette tableau il existe des Δ_{ij} qui sont supérieur à 0 donc on n'a pas atteint l'optimal donc le payson ne fait pas la bonne formulation. (0,5)

(2) Pour trouver l'optimal on va suivre l'algorithme des potentiel.

$Z_1 = 280\times40 + 170\times10 + 350\times20 + 150\times20 + 200\times40 = 30900$

$\theta_1 = \min(280, 200, 350) = 200$

	GE	EO	DE	a_i	u_i
HK	40 − 80	40 + 10	10 370	450	0
TA	10 + 200	20 − 150	30 −50	350	−30
RA	50 −40	20 350	40 −60	350	−30
b_j	280	500	370		1150
v_j	40	50	10		

(2)

$Z_2 = 80\times40 + 370\times10 + 200\times10 + 150\times20 + 350\times20 = 18900$

$\theta_2 = \min(80, 150) = 80$

	GE	EO	DE	a_i	u_i
HK	40 −10	40 80	10 370	450	0
TA	10 280	20 70	30 −40	350	−20
RA	50 −40	20 350	40 −50	350	−20
b_j	280	500	370		1150
v_j	30	40	10		

(1)

$Z_3 = 80\times40 + 370\times10 + 280\times10 + 70\times20 + 350\times20 = 18100$

tous les $\Delta_{ij} \leq 0$ alors on a atteint l'optimum avec $Z^* = 18100$

$x_{12} = 80$
$x_{13} = 370$
$x_{21} = 280$
$x_{22} = 70$
$x_{32} = 350$

$x_{11} = x_{23} = x_{31} = x_{33} = 0$

(0,5)

University Echahid Hamma Lakhdar — Exam of English

3rd year undergraduate

First and Family Name:...../..... +2

Groupe:...../.....

Correction of Exam Semester 4

Exercise01:3pts — Human Developing

When early humans hunted and gathered food, they were not in control of their environment. They could only interact with their surroundings as lower organisms did. When humans learned to make fire, however, they became capable of altering their environment. To provide themselves with fuel they stripped bark from trees, causing the trees to die. Clearings were burned in forests to increase the growth of grass and to provide a greater grazing area for the wild animals that humans fed upon. This development led to farming and the domestication of animals. Fire also provided the means for cooking plants which had previously been inedible. Only when the process of meeting the basic need for food reached a certain level of sophistication was it possible for humans to follow other pursuits such as the founding of cities.

Reading passage

1. This passage is mainly concerned with ----

- A) the evolution of farming techniques
- B) the role of hunting as a source of food
- C) how the discovery of fire changed the development of mankind
- D) basic food-gathering techniques of early humans
- E) how people supplied themselves with food prior to the discovery of how to make fire

2. One can infer from the passage that the discovery of how to make fire ----.

- A) improved the hunting skills of early humans
- B) caused early humans to interact with their surroundings as lower organisms did
- C) taught early humans how to live with lower organisms
- D) increased dietary options for early humans
- E) made easier for early humans to gather food

Good luck

3. As we understand from the passage, early humans ----

- A) didn't eat plants before they learned how to control fire
- B) used fire as a tool to alter their surroundings
- C) gained better control of their environment when they learned to live with lower organisms (1)
- D) started to maintain their food supply by hunting and gathering food when they started cooking with fire
- E) were the prey of many predators

Exercise02:5pts

Choose the appropriate options to complete the sentences:

Couch potatoes	wrestle	Rush hour	subject	believe

1. My favorite ---- at school are Math and English. (1) Subjects
2. I don't --- that kind of behavior in my classes. Please do not do it again. (1) believe
3. My parents are ---. They spend most of their time in front of TV. They never do exercise or other activities. (1) Couch potatoes
4. You are two close friends. Please do not ---- about such small things. (1) wrestle
5. Do you hate driving in this evening ----? Of course I do. (1) Rush hour

Exercise03(2.5pts): Complete the definitions of the nouns.

1. A ruler is something [b] a. which looks like a piano. (2.5)
2. A referee is a person [e] b. which helps us measure lengths or draw straight lines. (2.5)
3. An orphan is a child [d] c. which is rich in vitamin C. (0.5)
4. A cabbage is a vegetable [c] d. who has lost his parents by birth. (0.5)
5. An organ is a musical instrument [a] e. who controls a sports match or contest. (0.5)

xercise04: 5pts

atch the definition with the words in the box

adventure	appointment	hiking	poem	archer

1. Making a long journey on foot
Hiking
2. A kind of literary work written in short lines
(1) Poem
3. A prior arrangement to meet
(1) Appointement
4. A dangerous but exciting activity
(1) Adventure
5. A kind of sport practiced with a bow and arrows
(1) Archery

Exercise05: (5pts) Give the correct definition in the box from the table

dynamic	elegant	obedient	obstinate	tolerance

1. I don't expect him to change his mind because I know he is very ----.
(1) Obstinate
2. If you want to shop for the latest fashions or expensive souvenirs in New York City, go to Fifth Avenue. It is full of---- shops
(1) Elegant
3. Our teacher is a(an) ---- person, so she easily captivates the interest and attention of the students while she is teaching.
(1) Dynamic
4. My father gets angry with us whenever we make a mistake. He has no ---- for mistakes.
(1) Tolerance
5. Teachers like ---- students who never break their rules.
(1) Obedient

Université d'El-Oued
Département d'informatique
Chargée de module : Chourouk Guettas

Niveau : 3 LMD
Année : 2016-2017
Durée : 01:30 h

Examen de Génie Logiciel - corrigé_type

Question de cours : (7 points)

1) Citez deux objectifs du GL ? **(1 point)**
   - Fiabilité du logiciel
   - Contrôler le temps
   - Contrôler le coût
   - ...

2) Citez deux avantages de l'approche des incréments.
   - Bonne gestion de l'équipe (gain du temps) **(0.5 point)**
   - Livraison parallèle **(0.5 point)**

3) C'est quoi l'objectif de l'activité de test ? Quelle est la limite de l'activité de test ? Comment on peut surmonter ce problème ? Citer une approche en GL dont l'objectif principal est de vaincre la limite de test. **(2 point)**

Découvrir les erreurs	On peut rater des erreurs
La preuve de correction	Transformation formelle

4) Pourquoi on utilise les classes abstraites dans la programmation OO ?

Une classe abstraite sert de base à d'autres classes dérivées (héritées). Elle permet de définir des méthodes dont l'implémentation se fait dans les classes filles, alors elle sert essentiellement à créer un nouveau type d'objets. **(1 point)**

5) Compléter la table : **(2 points)**

Approches de développement	Type de Système
Approche de transformation formelle	Système critique
Approche de prototypage	Système de simulation graphique
Approche de réutilisabilité	Système d'informations
Approche de la cascade	Système d'information

Exercice 1: (6 points)

```
public class Employe{
    private String Nom;
    private String Prenom;
    private Date DateNaissance;
    private Date DateEmbauche;
    private float Quotite;
    private Fonction fonction_ ;
    private Service service_ ;
    private List<Salaire> salaire_ ;
    private List<Congé> congé_ ;

    Employe(String Nom, String prenom, Date datNais, Date datEm, float quotite){
```

```
                this.Nom = nom ;
                this.Prenon = prenom ;
                this.DateNaissance = datNais ;
                this.DateEmbauche = datEm ;
                this.Quotite = quotite ;
                fonction_ = new Fonction() ;
                service_ = new Service() ;
                salaire_ = new ArrayList<Salaire>();
                conge_ = new ArrayList<Conge>();
        }
        public int Age() {}
        public int CongésRestant() {}
        public int CongésTotaux(){}
        public int CongésPris(){}
        public void PrendreCongés(Date Début, Date Fin) {}
        public void ChangerSalaire(float B, float P, float S) {}
    }

public class Fonction {
        private String Nom;
        private Int RTT;
        Fonction() {}
}

public class Service {
        private String Nom;
        private Int RTT;

        Service() {};
}

public class Conge {
        private Date DateDebut;
        private Date DateFin;

        public Conge() { }
        public Int NbJours() {}
}

public class Salaire {
        private float ChargesPatronales;
        private float ChargesSalaires;
        private float Brut;
        private Date DateDebut;
        private Date DateFin;

        public Salaire() { }
        public float SalaireCharge() {   }
        public float SalaireNet() { }
        public Boolean SalaireEnCours() {  }
}
```

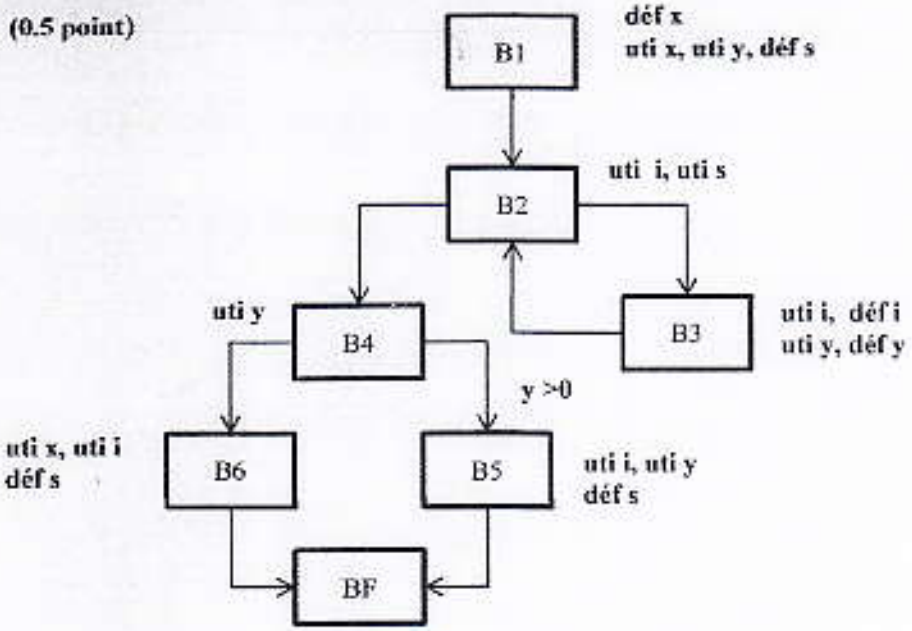
Exercice 2: (Activité de test) (7 points)

1) Soit le fragment du code suivant :

```
Read(x)
s = x * y
while (i < s) do
Begin
        i = i + 1
        y = y -1
End
if (y > 0) then s = y*i
else s = x + i
```

a) Etablir un test pour détecter les anomalies concernant les variables **y** et **i** de ce code. Après cette analyse, proposer une correction.

(0.5 point)



Les chemins possibles d'exécutions du programme (4 chemins possibles) (0.5 point)

C1: B1, B2, B4, B6, BF
C2: B1, B2, B4, B5, BF
C3: B1, B2, B3, B2, B4, B6, BF
C4: B1, B2, B3, B2, B4, B5, BF

variable y: (0.5 point)

C1: uti y, uti y **(anomlie utilisation sans definition)**
C2: uti y, uti y, uti y **(anomlie utilisation sans definition)**
C3: uti y, uti y, uti y **(anomlie utilisation sans definition)**
C4 : uti y, uti y, uti y, uti y **(anomlie utilisation sans definition)**

variable i: (0.5 point)

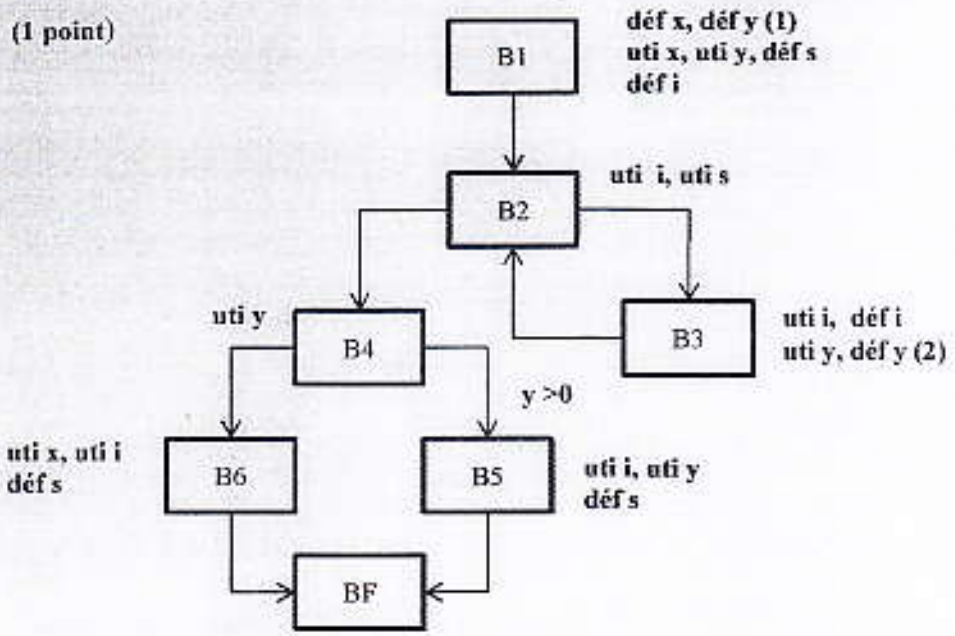
C1: uti i, uti i **(anomlie utilisation sans definition)**

C2: uti i, uti i **(anomlie utilisation sans definition)**
C3: uti i, uti i, def i, uti i **(anomlie utilisation sans definition)**
C4 : uti i, uti i, def i, uti i **(anomlie utilisation sans definition)**

Une correction dans laquelle les variables y et i seront définis (0.5 point)

```
Read(x)
Read(y)
s = x * y
i = 0
while (i < s) do
Begin
        i = i + 1
        y = y -1
End
if (y > 0) then s = y*i
else s = x + i
```

(1 point)

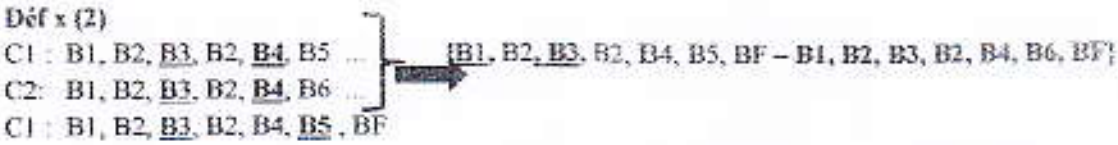


b) Etablir un test dynamique avec deux couvertures (couverture de toutes les instructions, couverture de toutes les utilisations du variable y).

Couverture de toutes les instructions (autres couvertures sont possibles) (0.5 point)
Couverture = { B1, B2, B3, B2, B4, B6, BF
B1, B2, B4, B5, BF}

Couverture de toutes les utilisations du variable y : (2 points)
Déf x (1)
C1 : **B1, B2** ...
C2 : B1, B2, **B3**, B2 ....
C3: B1, B2, **B4**, B5 ...
C4: B1, B2, **B4**, B6,BF
C5: B1, B2, ... **B5**, BF

⟹ **{B1, B2, B3,** B2, B4, B5, BF – **B1, B2, B4, B6, BF}**



c) Créer deux mutants de ce fragment. **(1 point)**

Mutant 1 :

```
Read(x)
Read(y)
s = x * y
i = 0
while (i >= s) do
Begin
        i = i + 1
        y = y -1
End
if (y > 0) then s = y*i
else s = x + i
```

Mutant 2 :

```
Read(x)
Read(y)
s = x * y
i = 0
while (i < s) do
Begin
        i = i + 1
        y = y -1
End
if (y <= 0) then s = y*i
else s = x + i
```