

Examen 01

Exercice 01 : Structure d'un algorithme

1. Définir c'est quoi un algorithme
2. Donner la structure de base d'un algorithme.
3. Donner la syntaxe des instructions suivantes :
 - Instruction d'affectation
 - Instruction d'entrée
 - Instruction de sortie.

Exercice 02 : Sémantique des instructions algorithmiques

Donner le sens (sémantique) des instructions algorithmiques suivantes :

	Instructions
1	Variables h : chaîne i : logique j : réel m : chaîne*20 k, i : entier
2	Variables note : tableau[1 .. 20] de réels t : tableau[1 .. 10, 1..5] d'entiers
3	Pour i ← 5 jusqu'à 13 pas 2 Ecrire("Salut ", i) FinPour
4	Si note > 20 ou note < 0 alors Ecrire("Note") Ecrire("\n") Ecrire("Incorrecte") FinSi
5	Structure personne nom : chaîne*20

	prenom : chaîne*15 age : entier EndStruct Variable p : personne
--	--

Exercice 03 : Surface d'un rectangle

Ecrire un algorithme qui calcul la surface d'un rectangle de longueur A et de largeur B saisies au clavier.

Exercice 04 : Interprétation et appel d'un sous algorithme

1. Que fait la fonction suivante :

```

Fonction Fonction1(n : entier, m : entier) : entier
Variables i, resultat : entiers
Debut
  resultat ← 0
  Pour i ← n jusqu'à m faire
    resultat ← resultat + i
  FinPour
  Retourner resultat
FinFonct
  
```

2. Ecrire un algorithme qui appelle et intègre cette fonction

Exercice 05 : Affichage à l'écran

Qu'affiche cette partie de l'algorithme à l'écran. Justifier votre réponse :

```

Pour i ← 1 jusqu'à 10 faire
  Ecrire("Bonjour ", i)
  Si i = 2 alors
    i ← i + 8
  FinSi
FinPour
i ← i + 2
Ecrire("La valeur de i est : ", i)
  
```

Exercice 06 : Equation du second degré



1. Ecrire un algorithme qui résout l'équation du second degré suivante : $A * x^2 + B * x + C = 0$. On suppose que A est différent de 0.

Exercice 07 : Fusion de deux tableaux

1. Soit T1 et T2 deux tableaux. Ecrire un algorithme qui réalise la fusion de T1 et T2 dans un tableau T3.
2. Refaire l'algorithme en fusionnant T1 et T2 dans T1.

Exercice 08 : Comparaison de trois nombres

Ecrire un algorithme qui compare trois nombre réels différents A, B et C.

Exercice 09 : Produit de deux matrices

Ecrire algorithme qui calcule la matrice C qui est le produit de deux matrices A et B. A est de l'ordre $n*m$ et B est de l'ordre $m*k$.

Exercice 10 : Calcul de la combinaison C_n^p

1. Ecrire un sous algorithme qui calcule la factorielle d'un nombre entier positif en utilisant une fonction récursive.
2. Ecrire un algorithme qui calcule la combinaison statistique $C(p,n)$:

$$C(p,n) = C_n^p = \frac{n!}{p!(n-p)!}$$



Correction de l'examen 01

Exercice 01 : Structure d'un algorithme

1. Définition d'un algorithme.

Un algorithme est la description de la méthode de résolution d'un problème quelconque en utilisant des instructions élémentaires. Ces instructions élémentaires deviennent compréhensibles par l'ordinateur lors de la traduction de l'algorithme en un programme.

2. Structure de base d'un algorithme.

Entête	{	<i>Algorithme</i> NomAlgorithme
Partie déclarative	{	<i>Constante</i> Identificateur = valeur <i>Variable</i> Identificateur : type
Corps de l'algorithme	{	Début Instruction 1 Instruction 2 Instruction n Fin

3. Syntaxe de l'affectation, instruction d'entrée et de sortie :

Syntaxe de l'affectation

Variable \leftarrow Expression

Expression peut être soit :

- Identificateur
- Constante
- Expression arithmétique
- Expression logique

Syntaxe de l'instruction d'entrée

Algorithmique

Lire (identificateur1, identificateur2, ..., identificateurN)

Syntaxe de l'instruction de sortie

Ecrire(Expression)

Expression peut être une valeur, un résultat, un message, le contenu d'une variable, etc.

Exercice 02 : Sémantique des instructions algorithmiques

Sémantique	
1	Déclaration de la variable <i>h</i> de type chaîne de caractères. Déclaration de la variable <i>i</i> de type logique. Déclaration de la variable <i>j</i> de type réel. Déclaration de la variable <i>m</i> de 20 caractères. Déclaration de deux variables <i>k</i> et <i>i</i> de type entiers.
2	Déclaration d'un tableau <i>note</i> de 20 éléments. Déclaration d'une matrice <i>t</i> de 10 lignes et 5 colonnes (tableau à deux dimensions).
3	Elle s'agit de la boucle <i>Pour</i> . Elle affiche à l'écran : Salut 5 Salut 7 Salut 9 Salut 11 Salut 13.
4	Elle s'agit de la structure alternative <i>SI ... Alors ... FinSI</i> . Si la valeur de variable <i>note</i> est supérieure à 20 ou inférieure à zéro, ces instructions affichent à l'écran : Note Incorrecte. L'instruction <i>Ecrire</i> ("\\n") réalise un retour à la ligne.
5	<i>personne</i> est une structure (nouveau type) qui contient trois champs qui sont : nom, prenom et age. <i>p</i> est une variable de type <i>personne</i> .

Exercice 03 : Surface d'un rectangle

Algorithme

Algorithme surface_rectangle

// A: longueur, B : largeur et S : surface

Variable A, B, S : réel

Examens Corrigés

Début

Ecrire("Saisir la longueur A : ")

Lire(A)

Ecrire("Saisir la largeur B : ")

Lire(B)

// Calcul de la surface du rectangle

$S \leftarrow A * B$

// Affichage du résultat

Ecrire("La surface du rectangle est : ", S)

Fin

Exercice 04 : Interprétation et appel d'un sous algorithme

1. La fonction1 renvoie la somme des entiers compris entre *n* et *m* (inclus).
2. Voilà un algorithme qui appelle et intègre la fonction1 :

Algorithme Appel_Fonction1

Vraibales *n, m* : entiers

' Déclaration de la fonction1

Fonction Fonction1(*n* : entier, *m* : entier) : entier

Variables *i, resultat* : entiers

Debut

resultat \leftarrow 0

Pour *i* \leftarrow *n* jusqu'à *m* faire

resultat \leftarrow resultat + *i*

FinPour

Retourner resultat

FinFonct

' Programme principal

Debut

Ecrire("Donnez la valeur de *n*")

Lire(*n*)

Ecrire("Donnez la valeur de *m*")

Lire(*m*)

' Appel de la fonction1

Ecrire("Fonction1(*n, m*)", Fonction1(*n, m*))



Algorithmique

Fin

Remarque :

La fonction1 renvoie 0 si $n > m$ car les instructions suivantes affiche 0 si $n > m$:

```
resultat ← 0
Pour i ← n jusqu'à m faire
    resultat ← resultat + i
FinPour
Ecrire(resultat)
```

**Exercice 05 : Affichage à l'écran**

1. Cette partie de l'algorithme affiche à l'écran :

```
Bonjour 1
Bonjour 2
La valeur de i est : 13
```

Remarque et justification :

Le message *Bonjour 10* ne sera pas affiché à l'écran car la valeur de i va subir deux incréments successives lorsque i vaut 2. La première est $i \leftarrow i + 8$ et qui donnera 10 et une autre incrémentation qui est automatique dû à la boucle *Pour*.

Donc juste après la boucle, la valeur de i est égale à 11.

L'instruction $i \leftarrow i + 2$ affectera à i la valeur 13, ce qui justifie l'affichage du message : La valeur de i est : 13

Exercice 06 : Equation du second degré

Pour résoudre une équation de second degré $A * x^2 + B * x + C = 0$, avec A différent de 0, nous calculons le discriminant (delta) :

$$\Delta = b^2 - 4ac$$

-si $\Delta = 0$, l'équation a une racine réelle double :

$$x_1 = -\frac{b}{2a}$$

- si $\Delta > 0$, l'équation a deux solutions réelles :

Examens Corrigés

$$x_1 = \frac{-b + \sqrt{\Delta}}{2a}$$

$$x_2 = \frac{-b - \sqrt{\Delta}}{2a}$$

- sinon l'équation n'admet pas de solutions dans l'ensemble R .

1. Algorithme

Algorithme equation_second_degré

Variable $X_1, X_2, A, B, C, \Delta$: réel

Début

Ecrire("Donnez les valeurs de A, B et C : ")

Lire(A, B, C)

// Calcul de delta

$\Delta \leftarrow B^2 - 4 * A * C$

Si $\Delta > 0$ alors

$X_1 \leftarrow (-B + \text{Sqrt}(\Delta)) / 2 * A$

$X_2 \leftarrow (-B - \text{Sqrt}(\Delta)) / 2 * A$

Ecrire("L'équation admet deux solutions $X_1 = "$, X_1 , " et $X_2 = "$, X_2)

Sinon Si $\Delta = 0$ alors

$X_1 \leftarrow -B / 2 * A$

Ecrire("L'équation admet une seule solution $X_1 = "$, X_1)

Sinon

Ecrire("L'équation n'admet pas de solutions dans l'ensemble R ")

FinSi

fin

Remarque :

La fonction *Sqrt(Delta)* calcule la racine carrée de Delta. En VB c'est la fonction *Sqr(Delta)*.

Remarque :

Si $A = 0$ alors l'équation du second degré $Ax^2 + Bx + C = 0$ deviendra une équation de premier degré, voir l'exercice 05 de l'atelier 02.

Exercice 07 : Fusion de deux tableaux

1. L'algorithme qui réalise la fusion de T_1 et T_2 dans un tableau T_3 .

On suppose que les tailles réelles des tableaux T_1 et T_2 sont inférieurs à 500.

Algorithmique

Algorithme Fusion_deux_tableaux

Variables T1 : tableau[1..500] de réels

T2 : tableau[1..500] de réels

T3 : tableau[1..1000] de réels

// N : taille réelle du tableau T1, $N \leq 500$

// M : taille réelle du tableau T2, $M \leq 500$

N, M, i : entier

Début

Ecrire("Donnez les tailles des tableaux T1 et T2 : ")

Lire(N,M)

// Remplissage du tableau T1

Pour i ← 1 jusqu'à N faire

Ecrire("Donnez T1[", i, "]")

Lire(T1[i])

FinPour

// Remplissage du tableau T2

Pour i ← 1 jusqu'à M faire

Ecrire("Donnez T2[", i, "]")

Lire(T2[i])

FinPour

// Fusion de T1 et T2 dans T3

Pour i ← 1 jusqu'à N + M faire

Si $i \leq N$ alors

T3[i] ← T1[i]

Sinon

T3[i] ← T2[i - N]

FinSi

FinPour

// Affichage du tableau T3

Pour i ← 1 jusqu'à M + N faire

Ecrire(" T3[", i, "] = ", T3[i])

FinPour

Fin

2. L'algorithme qui fusionne T1 et T2 dans T1.

Algorithme Fusion_des_tableaux_T1_et_T2_dans_T1

Variables T1 : tableau[1..500] de réels

Examens Corrigés

T2 : tableau[1..500] de réels

// N : taille réelle du tableau T1, $N \leq 500$

// M : taille réelle du tableau T2, $M \leq 500$

N, M, i : entier

Début

Ecrire("Donnez les tailles des tableaux T1 et T2")

Lire(N,M)

// Remplissage du tableau T1

Pour i ← 1 jusqu'à N faire

Ecrire("Donnez T1[", i, "]")

Lire(T1[i])

FinPour

// Remplissage du tableau T2

Pour i ← 1 jusqu'à M faire

Ecrire("Donnez T2[", i, "]")

Lire(T2[i])

FinPour

// Fusion de T1 et T2 dans T1

Pour i ← 1 jusqu'à N + M faire

Si $i \leq N$ alors

T1[i] ← T1[i]

Sinon

T1[i] ← T2[i - N]

FinSi

FinPour

// Affichage du tableau T1

Pour i ← 1 jusqu'à M + N faire

Ecrire(" T1[", i, "] = ", T1[i])

FinPour

Fin

Remarque :

La suite d'instructions :

// Fusion de T1 et T2 dans T1

Pour i ← 1 jusqu'à N + M faire

Si $i \leq N$ alors

T1[i] ← T1[i]



Algorithmique

```

Sinon
  T1[i] ← T2[i - N]
FinSi
FinPour

```

Peut être remplacés par :

```

// Fusion de T1 et T2 dans T1
Pour i ← 1 jusqu'à M faire
  T1[i + N] ← T2[i]
FinPour

```

Exercice 08 : Comparaison de trois nombres

Il y a six cas possible qui sont :

$A > B > C$; $A > C > B$; $B > A > C$; $B > C > A$; $C > B > A$; $C > A > B$.

Algorithme :

Algorithme comparaison_trois_nombres

Variable A, B, C : réel

Début

Ecrire("Donnez trois nombre différents; A, B et C : ")

Lire(A, B, C)

SI A > B Alors

SI B > C Alors

Ecrire(A, ">", B, ">", C)

Sinon

SI A > C Alors

Ecrire(A, ">", C, ">", B)

Sinon

Ecrire(C, ">", A, ">", B)

FinSi

FinSi

Sinon

SI A > C Alors

Ecrire(B, ">", A, ">", C)

Sinon

SI B > C Alors

Ecrire(B, ">", C, ">", A)

Examens Corrigés

```

Sinon
  Ecrire(C, ">", B, ">", A)
FinSi
FinSi
FinSi
Fin

```

Exercice 09 : Produit de deux matrices

Voir l'exercice 06 de l'atelier 03 (Opérations sur les tableaux).

Exercice 10 : Calcul de la combinaison C_n^p

1. Voir l'exercice 10 de l'atelier 06 (fonctions et procédures).
2. Algorithme qui calcule la combinaison statistique :

Algorithme combinaison

Variable p, n : entier

// Déclaration de la fonction Fact qui calcule la factorielle

Fonction Fact (m : entier) : entier

Début

Si m = 0 Alors

Retourner 0

Sinon

Retourner Fact(m-1) * m

FinSi

FinFonction

// Algorithme principal

Début

Ecrire (" Donner un entier p positif : ")

Lire(p)

Ecrire(" Donner un entier n positif : ")

Lire(n)

// Calcul de la combinaison C(p,n)

Ecrire("La combinaison C(p,n) est : ", fact(n) / (fact(p) * fact(n - p))

Fin

