

Correction TD1

Chapitre 1 : Rappels et Introductions

Algorithme 2– SMI/S3

LIRE ()

ECRIRE ()



EXERCICE 1

► Énoncé:

Écrire un algorithme qui demande un nombre à l'utilisateur, puis affiche ce nombre à l'écran.

Éléments de réponse

- ▶ Un programme informatique est utile quand il peut interagir avec l'utilisateur, cette interaction se traduit de la façon suivante:
 - le programme affiche des messages à l'utilisateur: ECRIRE ()
 - L'utilisateur saisit des données à traiter par le programme: LIRE ()

Correction exercice 1

ALGORITHME exercice1

Variables

X : réel

DEBUT

 ECRIRE ("donner un nombre: ")

 Lire (X)

 ECRIRE (" le nombre saisi est :
 ", X)

FIN



L'exécution

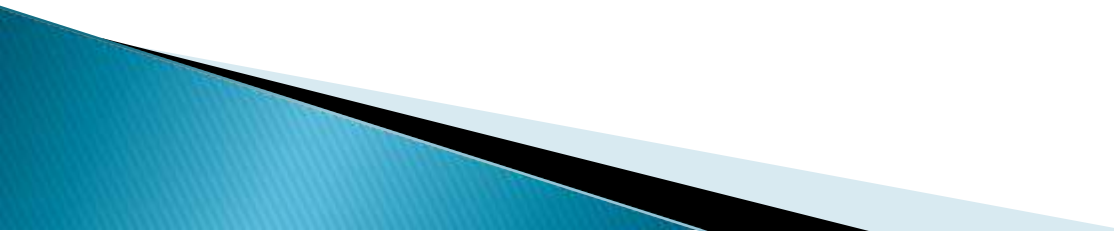
Instructions	Exécution
ECRIRE ("donner un nombre: ")	Saisir votre nombre :
LIRE (X)	Attendre la saisie
	L'utilisateur a saisi: 3 donc $x=3$
ECRIRE (" le nombre saisi est : ", X)	le nombre saisi est : 3

EXERCICE 2

► Énoncé:

Écrire un algorithme qui demande un nombre à l'utilisateur, puis calcule et affiche le carré de ce nombre.

Éléments de réponse

- Le programme affiche un message à l'utilisateur pour écrire un nombre: ECRIRE ()
 - L'utilisateur saisit un nombre à traiter par le programme: LIRE ()
 - Le programme calcule le carré
 - Le programme affiche le résultat
- 

Algorithme

ALGORITHME exercice_2

Variables

X, R: REEL

DEBUT

 ECRIRE ("donner un nombre: ")

 LIRE (X)

$R \leftarrow X * X$

 ECRIRE("le carré de", X, "est: ", R)

FIN



Exécution

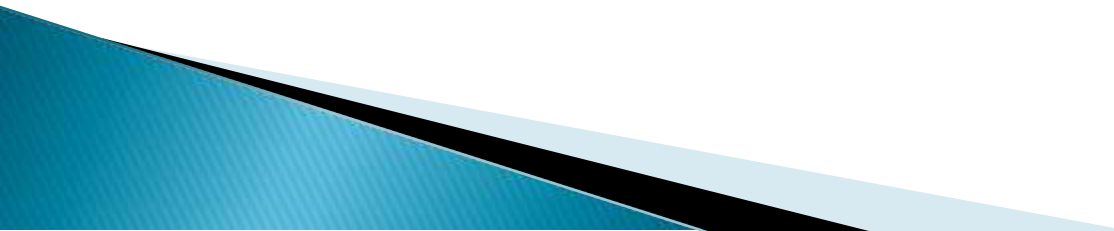
Instructions	Exécution
ECRIRE ("donner un nombre: ")	saisir le nombre
LIRE (X)	donner la main pour écrire la valeur de X
	L'utilisateur a saisi: 3 donc $X = 3$
$R \leftarrow X * X$	Calcule le carré et affecte au variable R le résultat
ECRIRE("le carré de", X, "est: ", R)	Le carré de 3 est 9

Exercice 3

- ▶ **Enoncé:**

Écrire un algorithme qui demande deux nombres à l'utilisateur, puis calcule et affiche la somme de ces deux nombres.

Éléments de réponse

- Le programme affiche un message à l'utilisateur pour saisir deux nombres:
ECRIRE ()
 - L'utilisateur saisit deux nombres à traiter par le programme: LIRE ()
 - Le programme calcule la somme
 - Le programme affiche le résultat
- 

Algorithme

ALGORITHME exercice_3

Variables

A, B, R : réel

DEBUT

ECRIRE ("donner premier nombre: ")

LIRE (A);

ECRIRE ("donner deuxième nombre")

LIRE (B);

R<--- A+B;

ECRIRE ("la somme des deux nombres est: ", R)

FIN

ALGORITHME exercice_3

Variables

X1, X2, X3 : réel

DEBUT

ECRIRE ("donner deux nombres: ")

LIRE (A, B);

R<--- A+B;

ECRIRE ("la somme des deux nombres
est: ", R)

FIN

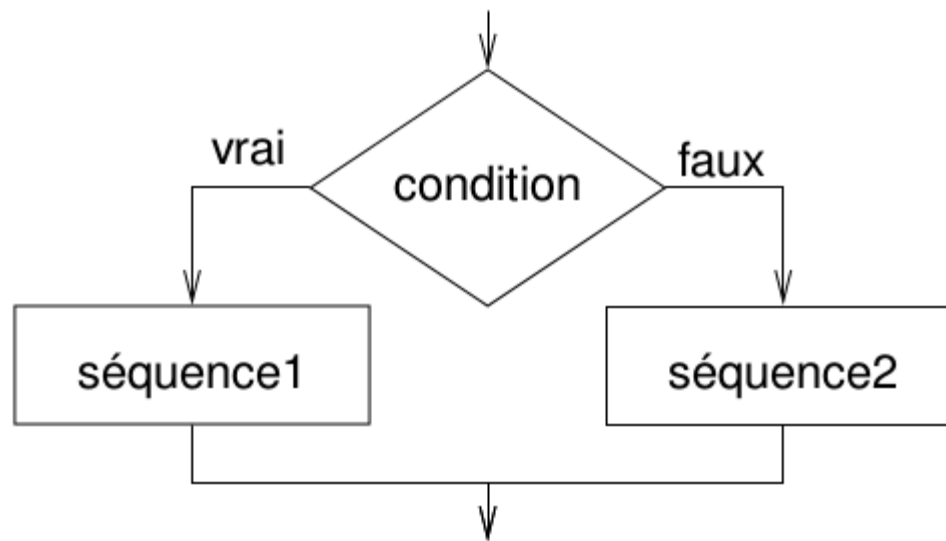
Exécution

Instructions	Exécution
ECRIRE ("donner premier nombre: ")	Entrer le 1 ^{er} nombre
LIRE (A)	donne la main pour écrire et il stocke la valeur de A
	L'utilisateur a saisi: 3 donc $A = 3$
ECRIRE ("donner deuxième nombre")	Entrer le 2 ^{ème} nombre
LIRE (B)	donne la main pour écrire et il stocke la valeur de B
	'utilisateur a saisi: 3 donc $B = 3$
$R \leftarrow A + B$	Calcule la somme et affecte au variable R le résultat
ECRIRE ("la somme des deux nombres est: ", R)	la somme des deux nombres est 5

Structures de contrôle



Conditionnelles



Exercice 6

- ▶ **Énoncé:**

Écrire un algorithme qui demande un nombre à l'utilisateur, puis calcule et affiche la parité de ce nombre (pair ou impair).

Éléments de réponse

→ Le programme évalue la parité → le programme traite différent cas de figure donc on aura besoin d'un traitement conditionnelle:

```
Si < condition > ALORS
    Traitement 1
Sinon
    Traitement 2
FinSi
```

→ On peut utiliser deux structures conditionnelles indépendantes:

```
si < condition > alors Traitement 1 finsi
```

```
si < condition > alors Traitement 2 finsi
```

Le résultat est le même mais le programme dans ce cas n'est pas optimisé

Algorithme

ALGORITHME exercice_6

Variable

A: entier

DEBUT

ECRIRE("donner un nombre: ");

LIRE (A);

SI $A \bmod 2 = 0$ ALORS

ECRIRE ("le nombre ", A, " est pair ")

SINON

ECRIRE ("le nombre ", A, " est impair ")

FINSI

FIN



Exécution

Instructions	Exécution
ECRIRE("donner un nombre: ");	Affichage de : donner un nombre
LIRE (A);	donne la main pour écrire et il stocke la valeur de A
	L'utilisateur a saisi: 3 donc $A = 3$
SI $A \bmod 2 = 0$ ALORS ECRIRE ("le nombre ", A, " est pair ") SINON ECRIRE ("le nombre ", A, " est impair ") FINSI	le nombre 3 est impair

Exercice 5

- ▶ **Enoncé:**

Écrire un algorithme qui demande trois nombres à l'utilisateur, puis affiche le plus grand nombre.

Éléments de réponse

→ Le programme évalue le maximum de trois nombres → le programme traite différent cas de figure donc on aura besoin d'un traitement conditionnelle , on peut utiliser par exemple « selon » pour traiter que les cas vrais et ignore les autres ce qui rend le programme plus claire et optimisé

Selon <sélecteur> faire

1: traitement 1

..

N: traitement N

→ On peut utiliser deux structures conditionnelles indépendantes:

si < condition > alors Traitement 1 fin si

si < condition > alors Traitement n fin si

Le résultat est le même mais le programme dans ce cas n'est pas optimisé

Algorithme

ALGORITHME exercice5

Var A, B, C, max: Réel

DEBUT

```
ECRIRE ("Donner le premier nombre: ");  
LIRE (A);  
ECRIRE ("Donner le deuxième nombre: ");  
LIRE (B);  
ECRIRE ("Donner le troisième nombre: ");  
LIRE (C);  
max<---A  
  SI B>max ALORS  
    max<---B  
  FINSI  
  SI C>max ALORS  
    max<---C  
  FINSI  
ECRIRE ("maximum est: ", max);
```

FIN

ALGORITHME exercice5

Variables:

A, B, C: Réel

DEBUT

```
ECRIRE ("Donner les trois nombres: ");  
LIRE (A, B, C);
```

selon A,B,C faire

A>B et A>C: ECRIRE ("le nombre max est:", A)

B>A et B>C: ECRIRE ("le nombre max est:", B)

C>A et C>B: ECRIRE ("le nombre max est:", C)

FINselon

FIN

Exécution

Instructions	Exécution
ECRIRE("donner un nombre: ");	Affichage de : donner un nombre
LIRE (A);	donne la main pour écrire et il stocke la valeur de A
	L'utilisateur a saisi: 6, 8, 4
selon A,B,C faire A>B et A>C: ECRIRE ("le nombre max est:", A) B>A et B>C: ECRIRE ("le nombre max est:", B) C>A et C>B: ECRIRE ("le nombre max est:", C) FINselon	le nombre max est: 8

Exercice 7

► Énoncé:

Écrire un algorithme permettant d'afficher la saison en introduisant le numéro du mois.

Les saisons :

De 12 à 2 : hiver

De 3 à 5 : printemps

De 6 à 8 : été

De 9 à 11 : automne



Éléments de réponse

→ vu que les mois qui représentent l'hiver sont 12, 1, 2 ce qui demande plusieurs expressions logiques pour exprimer la condition donc il est judicieux d'utiliser la structure conditionnelle à choix multiple:

Selon <sélecteur> faire

1: traitement 1

..

N: traitement N

Et dans ce cas on peut grouper les valeurs 12, 1, 2 pour un seul traitement

- ▶ Pour éviter que l'utilisateur entre un nombre différent des mois d'année par exemple 13, l'ordinateur peut prévenir l'utilisateur de réécrire un nombre entre 1 et 12, donc on va faire un traitement à l'aide de la boucle « Répéter »
- N.B: la structure conditionnelle indépendante donne le même résultat mais dans ce cas l'algorithme n'est pas optimisé car la machine évalue l'ensemble des conditions quelque soit l'exactitude de chaque condition

Algorithme

ALGORITHME exercice_7

Variables

A : entier

Début

Ecrire(" Donner un numéro de mois
de 1 à 12")

Répéter Lire(A)

Jusqu'à $A > 0$ et $A < 13$

Selon A faire

3,4,5 : Ecrire("La saison est :
PRINTEMPS")

6,7,8 : Ecrire("La saison est : ETE")

9,10,11 : Ecrire("La saison est :
AUTOMNE")

12,1,2 : Ecrire("La saison est : HIVER")

FinSelon

Fin

ALGORITHME exercice_7

variables

A: entier

DEBUT

ECRIRE ('donner le numéro de mois entre 1 et 12');
REPETER LIRE (A);
jusqu'à $A > 0$ et $A < 13$;

SI $A \geq 3$ et $A \leq 5$ ALORS

ECRIRE(' la saison est printemps');

FINSI

SI $A \geq 6$ et $A \leq 8$

ECRIRE(' la saison est l'été');

FINSI

SI $A \geq 9$ et $A \leq 11$ ALORS

ECRIRE(' la saison est l'automne');

FINSI

SI $A = 12$ ou $(A \leq 2$ et $A < 0)$

ECRIRE(' la saison est l'hiver');

FINSI

FIN

(algorithme proposé par le groupe 3)

Exécution

Instructions	Exécution
Ecrire(" Donner un numéro de mois de 1 à 12")	Affichage: Donner un numéro de mois 1 -- 12
Répéter Lire(A) Jusqu'à $A > 0$ et $A < 13$	Donner la main pour écrire le mois et évite que l'utilisateur entre un nombre différent des mois d'année par exemple 13, l'ordinateur n'accepte que les nombres entre 1 et 12,
Selon A faire 3,4,5 : Ecrire("La saison est : PRINTEMPS") 6,7,8 : Ecrire("La saison est : ETE") 9,10,11 : Ecrire("La saison est : AUTOMNE") 12,1,2 : Ecrire("La saison est : HIVER")	Si $A=3$ ou $A=4$ ou $A=5$: Ecrire('La saison est : PRINTEMPS') ; Si $A=6$ ou $A=7$ ou $A=8$: Ecrire('La saison est : ETE') ; Si $A=9$ ou $A=10$ ou $A=11$: Ecrire('La saison est : AUTOMNE') ; Si $A=12$ ou $A=1$ ou $A=2$: Ecrire('La saison est : HIVER') ;

Exercice 9

► Enoncé:

Compte tenu des cotés d'un triangle, écrire un algorithme qui permet de déterminer son type

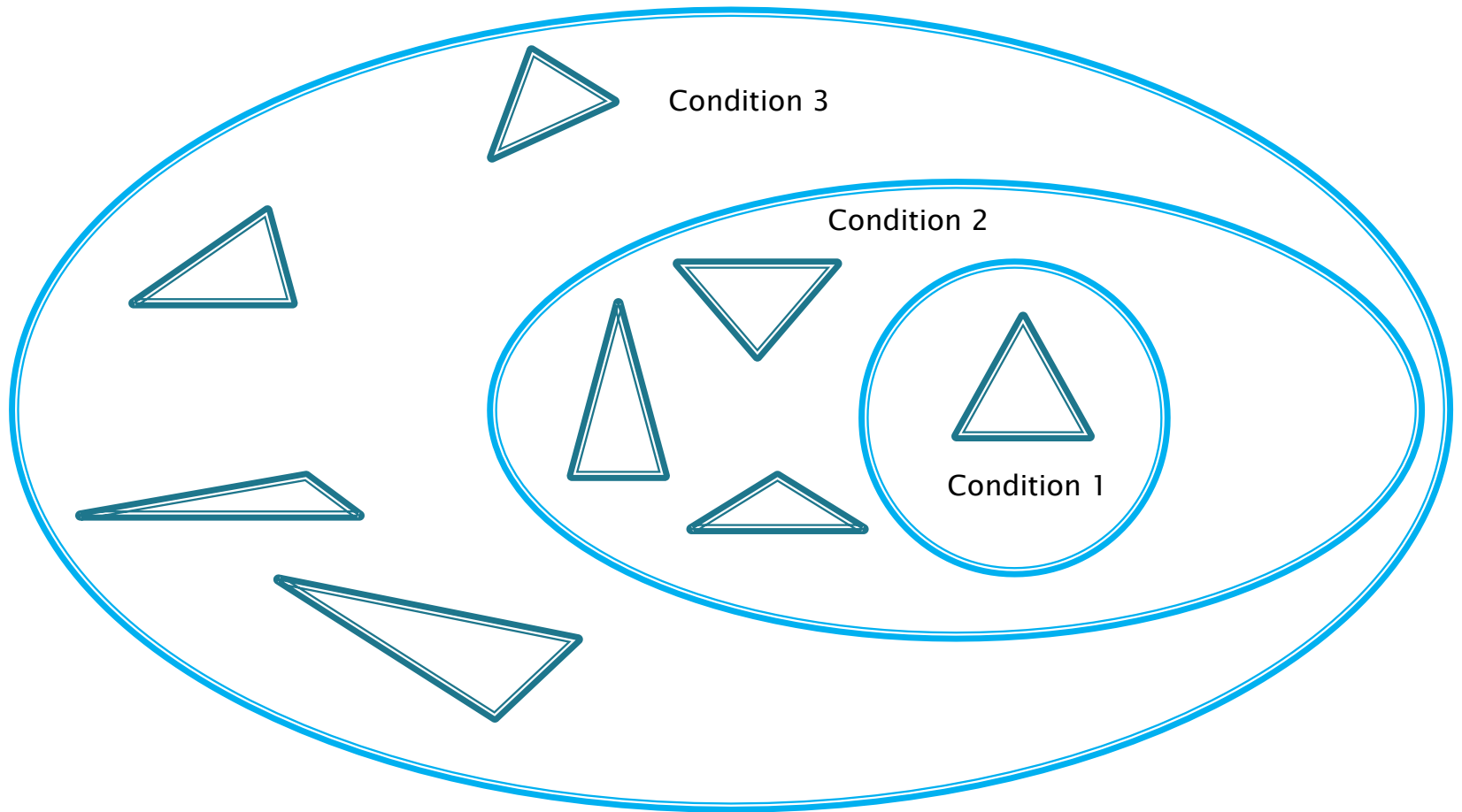
Types de triangle:

Équilatéral : les 3 cotés de mêmes longueurs

Scalène : tous ces cotés sont de longueurs différents

Isocèle : deux cotés de mêmes longueurs





Éléments de réponse

→ Le programme traite trois conditions → on aura besoin d'un traitement conditionnelle, on peut utiliser la structure conditionnelle imbriquée puisqu'on ne peut pas prédire l'exactitude de la deuxième condition si la première condition est fausse

→ donc il faut prévoir une autre condition dans le bloc « Sinon »

```
Si <condition1 > Alors
    Traitement 1
Sinon Si <condition2 > Alors
    Traitement 2
Sinon Traitement 2
    Fin si
Finsi
```

→ notez: on doit déclarer autant de FinSi que de Si

Algorithme

ALGORITHME exercice_9

Var X1, X2, X3: entier

DEBUT

ECRIRE ("donner premier cote: ");

LIRE (X1);

ECRIRE ("donner deuxième cote: ");

LIRE (X2);

ECRIRE ("donner troisième cote: ");

LIRE (X3);

SI $X1 == X2$ et $X2 == X3$ ALORS

 ECRIRE ("équilatéral");

 SINON SI $X1 <> X2$ et $X1 <> X3$ et $X2 <> X3$ Alors

 ECRIRE ("scalène");

 SINON

 ECRIRE ("isocèle");

 FINSI

FINSI

FIN

Les structures itératives



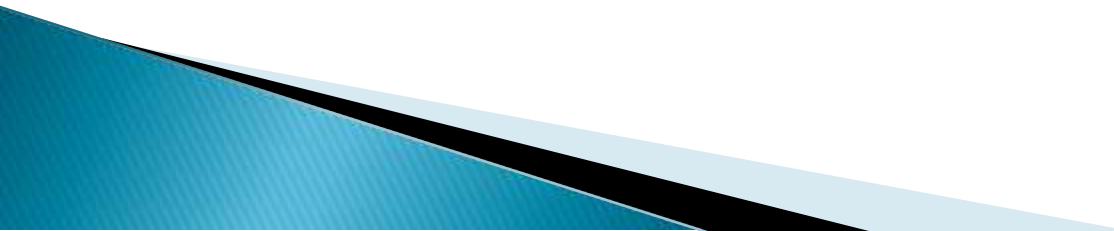
Exercie 4

- ▶ Énoncé:

Écrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres pairs suivants.



Éléments de réponse

- ▶ Le programme doit répéter le traitement de la parité plusieurs fois donc on aura besoin d'une boucle « structure itérative »
 - Le programme compte jusqu'à 10 nombres pairs après un nombre d'entrée saisi par l'utilisateur → vu que le nombre de départ est inconnu donc le nombre d'itération est inconnu aussi et la structure itérative utilisée sera « Tant que » .
 - La valeur d'entrée s'incrémentera et le programme traitera la parité, si la nouvelle valeur est pair le compteur de notre boucle s'incrémentera
- 

Algorithme

ALGORITHME exercice_4

Var X, N: entier

N ← 0

DEBUT

ECRIRE ("donner le premier nombre: ")

LIRE (X)

TANT QUE N < 10

X ← X + 1

SI X MOD 2 = 0 ALORS

ECRIRE (X)

N ← N + 1

FINSI

FinTantQue

Fin



Exécution

Instructions	Exécution
$N \leftarrow 0$	La valeur de N est initialisée à 0
ECRIRE ('donner le premier nombre: ');	L'ordinateur affiche la phrase : donner le premier nombre:
LIRE (X);	donne la main pour écrire et stocke la valeur de X
TANT QUE $N < 10$	L'ordinateur à la valeur de $N=0$ donc il va exécuter l'instruction de TANTQUE
$X \leftarrow X+1$;	L'ordinateur va incrémenter la valeur de X et il aura si X entrée est 3 alors la nouvelle $X=4$
SI $X \bmod 2 = 0$ ALORS	L'ordinateur vérifie la condition de SI 4 est pair donc il va exécuter l'instruction de Alors
ECRIRE (X);	Affichage de nombre 4
$N \leftarrow N+1$;	Incrémentation de N parce que la condition est vrai donc la valeur de $N=1$

Exercice 8

- ▶ Énoncé:

Écrire un algorithme permettant d'afficher les 15 premières valeurs de la série:

1, 1, 2, 3, 5, 8, 13, 21, ...

Élément de réponse

0	1	1	2	3	5	8	13	21	35	56	91
---	---	---	---	---	---	---	----	----	----	----	----	------	------	------

C'est une Suite de Fibonacci, La suite est définie par :
 $F_0=0$, $F_1=1$, $F_n=F_{n-1} + F_{n-2}$ pour $n>1$

$l=0$

F_{n1} ↓	F_{n2} ↓	F_{n3} ↓							
0	1	$0+1=1$							

$l=1$

	F_{n1} ↓	F_{n2} ↓	F_{n3} ↓						
0	1	$0+1=1$	$1+1=2$						

$l=2$

		F_{n1} ↓	F_{n2} ↓	F_{n3} ↓					
0	1	$0+1=1$	$1+1=2$	$1+2=3$					

Éléments de réponse (suite)

- ▶ Le programme doit répéter le calcul de la somme. Donc, on aura besoin d'une boucle « structure itérative ».
- Le programme fait la somme 15 fois → vu que le nombre d'itération est connu on utilisera la boucle « Pour »
- À chaque itération, le programme calcule $F_n = F_{n-1} + F_{n-2}$

Algorithme

- ▶ ALGORITHME exercice8
- ▶ Var Fn1, Fn2, Fn3: entier
- ▶ Fn1=0;
- ▶ Fn2=1;
- ▶ DEBUT
- ▶ ECRIRE(Fn1);
- ▶ ECRIRE(Fn2);
- ▶ pour i de 0 à 15 FAIRE
 - ▶ Fn3=Fn1 +Fn2
 - ▶ ECRIRE(Fn3);
 - ▶ Fn1 <---Fn2;
 - ▶ Fn2 <---Fn3;
- ▶ FIN

Exécution

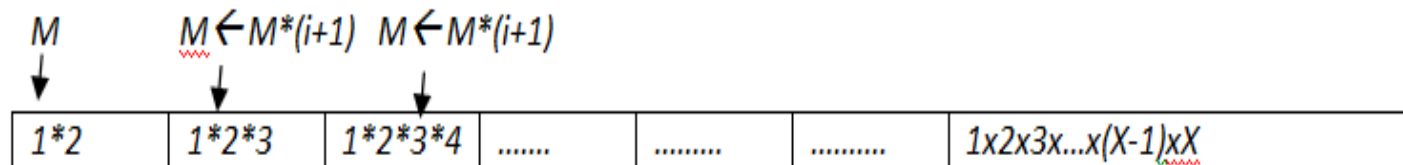
Instructions	Exécution
Fn1=0; Fn2=1;	Initialiser F0 et F1 , ces deux variable utilisé comme pointeur et on fiat une translation de ce pointeur à chaque fois que la boucle avance
pour i de 0 à 15 FAIRE	La boucle joue le rôle d'un compteur
Fn3=Fn1+Fn2	Pour i=0 l'ordinateur calcule la somme de Fn1 et Fn2 et l'affecte à un autre variable Fn3=0+1=1
ECRIRE(Fn3);	L'ordinateur affiche la valeur de Fn3
Fn1<---Fn2; Fn2<---Fn3;	Pour qu'on puisse calculer la nouvelle valeur de Fn3 l'ordinateur affecte la valeur de Fn2 à Fn1 → Fn1=1 et la valeur de Fn3 à Fn2 → 1
	l'ordinateur calcule la somme de Fn1 et Fn2 et l'affecte à un autre variable Fn3 dans ce cas on a les nouvelles valeurs suivantes : Fn1=1 Fn2=2 Donc Fn3=Fn1+Fn2=2

Exercice 10

► Énoncé:

Écrire un algorithme permettant de calculer la valeur de l'expression E, telle que:

$E = (1 \times 2) + (1 \times 2 \times 3) + (1 \times 2 \times 3 \times 4) + \dots + (1 \times 2 \times 3 \times \dots \times (X-1) \times X)$, et $(X \geq 2)$.



Éléments de réponse

- ▶ Vu que le nombre d'itération est connu on utilisera « Pour »
- ▶ La valeur saisie par l'utilisateur doivent être supérieure ou égale à 2. donc, on va filtrer les entrées par la structure itérative « Répéter »

Algorithme

algorithme exercice_10

Variables

N, M, X, i: entier

DEBUT

Répéter ECRIRE ("saisir le nombre X:")

LIRE (X)

jusqu'à $X \geq 2$;

M=2;

N=0;

POUR (i de 2 à X) Faire

$N \leftarrow N + M$

$M = M * (i + 1)$

FINPOUR

FIN

