

CORRECTION

Exercice I : On rappelle que :

$$x^1 = x$$

$$x^2 = x * x$$

$$x^3 = x * x * x$$

$$x^4 = x * x * x * x$$

...

$$x^{31} = x * x * x * \dots * x \text{ (31 fois)}$$

Ecrire un programme qui lit une valeur x quelconque et qui calcule x^{31} . On suppose qu'il n'existe pas de fonction permettant d'effectuer directement ce calcul. Il faut effectuer une suite de multiplications.

1. Imaginer un état intermédiaire

i	Puis
k	x^k

2. Voir comment progresser d'un état au suivant

i	Puis
k+1	x^{k+1}

Se fait par les instructions :

$$i \leftarrow i + 1$$

$$\text{Puis} \leftarrow \text{Puis} * x$$

3. Définir le critère de fin de boucle

Nous nous arrêtons quand $k = 31$.

4. Trouver comment commencer

Nous pouvons choisir l'état initial

I	Puis
1	x

D'après l'énoncé, la valeur x est fournie en donnée. L'état initial s'obtient donc par

$$i \leftarrow 1$$

Lire x

$$\text{Puis} \leftarrow x$$

Variables i, x, Puis : numériques

$$i \leftarrow 1$$

Lire x

$$\text{Puis} \leftarrow x$$

Répéter

$$i \leftarrow i + 1$$

$$\text{Puis} \leftarrow \text{Puis} * x$$

Jusqu'à i=31

Ecrire x, « élevé à la puissance 31 vaut », Puis

Autre solution : il est possible d'utiliser une boucle avec compteur.

Variables i, x, Puis : numériques

Lire x

$$\text{Puis} \leftarrow x$$

Répéter pour i=2 à 31

$$\text{Puis} \leftarrow \text{Puis} * x$$

Jusqu'à i=31

Ecrire x, « élevé à la puissance 31 vaut », Puis

Nous aurions pu choisir comme état initial :

I	Puis
0	1

Variables i , x , Puis : numériques

Lire x

Puis $\leftarrow 1$

Répéter pour $i=1$ à 31

Puis \leftarrow Puis * x

Fin Pour

Ecrire x , « élevé à la puissance 31 vaut », Puis

Exercice II : Ecrire un programme qui lit une valeur x quelconque et qui calcule la valeur de l'expressions

$$1 + x + x^2 + x^3 + x^4 + \dots + x^{20}$$

Il faut cumuler, dans la variable Exp, les différentes valeurs obtenues pour la variable Puis.

1. Imaginer un état intermédiaire

I	Puis	Exp
K	x^k	$1 + x + \dots + x^k$

2. Voir comment progresser d'un état au suivant

I	Puis	Exp
$k+1$	x^{k+1}	$1 + x + \dots + x^k + x^{k+1}$

Se fait par les instructions :

$i \leftarrow i+1$

Puis \leftarrow Puis * x

Exp \leftarrow Exp + Puis

3. Définir le critère de fin de boucle

Nous nous arrêtons quand $k = 20$.

4. Trouver comment commencer

Nous pouvons choisir l'état initial

i	Puis	Exp
1	x	$1 + x$

Variables i , x , Puis, Exp : numériques

$i \leftarrow 0$

Lire x

Puis $\leftarrow 1$

Exp $\leftarrow 1$

Répéter

$i \leftarrow i + 1$

Puis \leftarrow Puis * x

Exp \leftarrow Exp + Puis

Jusqu'à $i=20$

Ecrire « Pour $x=$ », x , « L'expression vaut », Exp

Comme précédent, nous pouvons employer une boucle avec compteur :

Variables i , x , Puis, Exp : numériques

Lire x

Puis $\leftarrow 1$

Exp $\leftarrow 1$

Répéter pour $i =1$ à 20

Puis \leftarrow Puis * x

Exp \leftarrow Exp + Puis

Fin Pour

Ecrire « Pour $x=$ », x , « L'expression vaut », Exp

Exercice III : Déterminez simultanément le maximum et le minimum de 50 valeurs fournies (on ne lit qu'une seule fois chaque valeur).

- a- En ne faisant aucune hypothèse sur les valeurs

1. Imaginer un état intermédiaire

i	Min	Max
k	Plus petite des k premières valeurs	Plus grande des k premières valeurs

2. Voir comment progresser d'un état au suivant

i	Min	Max
k+1	Plus petite des k+1 premières valeurs	Plus grande des k+1 premières valeurs

Se fait par les instructions :

```

i ← i+1
Lire Val
Si Val > Max alors Max ← Val Fsi
Si Val < Min alors Min ← Val Fsi
    
```

3. Définir le critère de fin de boucle

Nous nous arrêtons quand $k = 50$.

4. Trouver comment commencer

Nous pouvons choisir l'état initial

$k = 1$.

Il s'obtient par les instructions

```

i ← 1
Lire Val
Max ← Val
Min ← Val
    
```

Programme :

```

Variables i, Max, Min, Val : numériques
Lire Val
Max ← Val
Min ← Val
Répéter pour i = 1 à 50
    Lire Val
    Si Val > Max alors Max ← Val Fsi
    Si Val < Min alors Min ← Val Fsi
Fin Pour
Ecrire « Maximum », Max, « Minimum », Min
    
```

b- En supposant que le maximum est positif et que le minimum est négatif

Avec les hypothèses formulées, nous pouvons choisir comme état initial :

I	Min	Max
0	0	0

D'où le programme :

```

Variables i, Max, Min, Val : numériques
Max ← 0
Min ← 0
Répéter pour i = 1 à 50
    Lire Val
    Si Val > Max alors Max ← Val Fsi
    Si Val < Min alors Min ← Val Fsi
Fin Pour
Ecrire « Maximum », Max, « Minimum », Min
    
```

Exercice IV : Déterminez la plus grande valeur d'un nombre quelconque de valeurs non nulles. La dernière valeur sera 0.

Ici, il n'est plus nécessaire de compter les valeurs lues.

1. Imaginer un état intermédiaire

Un certain nombre de valeurs ont été lues et Max contient la plus grande

2. Voir comment progresser d'un état au suivant

Une valeur supplémentaire a été lue. Max contient la plus grande de toutes les valeurs déjà lues.

Se fait par les instructions :

 Lire Val

 Si Val > Max alors Max ← Val Fsi

3. Définir le critère de fin de boucle

Nous nous arrêtons quand Val = 0.

4. Trouver comment commencer

Nous pouvons choisir l'état initial :

- Une seule valeur lue

- Max contient cette valeur

Programme :

Variables Max, Val : numériques

Lire Val

Max ← Val

Répéter

 Lire Val

 Si Val > Max alors Max ← Val Fsi

Jusqu'à Val = 0

Ecrire « Maximum », Max

Cette solution pose problème si la première valeur entrée est 0. Il faut rentrer une deuxième valeur.

Solution modifiée

Variables Max, Val : numériques

Lire Val

Max ← Val

Si val = 0 alors ecrire « max », max

Sinon

 Répéter

 Lire Val

 Si (Val > Max et val <> 0) alors Max ← Val

Fsi

 Jusqu'à Val = 0

Ecrire « Maximum », Max

Fsi

Exercice V : Même chose que l'exercice précédent, mais en précisant en plus le « rang » de la plus grande valeur (deuxième, huitième, ...).

Il suffit de modifier le programme précédent. Il faut introduire :

- Une variable compteur i nous permettant d'attribuer un rang à chaque valeur lue.
- Une variable Rangmax qui va servir à déterminer le rang de la plus grande valeur.

1 Imaginer un état intermédiaire

I	Max	Rangmax
K	Plus grande des k premières valeurs	Rang de cette valeur (parmi les k premières)

2 Voir comment progresser d'un état au suivant

I	Max	Rangmax
---	-----	---------

k+1	Plus grande des k+1 premières valeurs	Rang de cette valeur (parmi les k +1 premières)
-----	---------------------------------------	---

Instructions :

$i \leftarrow i + 1$

Lire Val

Si Val > Max alors

$\text{Max} \leftarrow \text{Val}$

$\text{Rangmax} \leftarrow i$

Fsi

3 Définir le critère de fin de boucle

Nous nous arrêtons toujours quand Val = 0

4 Trouver comment commencer

Lire Val

$\text{Max} \leftarrow \text{Val}$

$\text{Rangmax} \leftarrow 1$

5 Nous pouvons choisir l'état initial :

Nous choisissons comme valeur initiale celui où une seule valeur a été lue (k=1)

Solution modifiée

Variables Cpt, RangM, Max, Val : numériques

Lire Val

$\text{Max} \leftarrow \text{Val}$

$\text{Cpt} \leftarrow 1$

$\text{RangM} \leftarrow 0$

Si val = 0 alors écrire « max », max, « rang max= », RangM

Sinon

Répéter

Lire Val

$\text{Cpt} \leftarrow \text{Cpt} + 1$

Si (Val > Max et Val <> 0) alors

$\text{Max} \leftarrow \text{Val}$

$\text{RangM} \leftarrow \text{Cpt}$

Fsi

Jusqu'à Val = 0

Ecrire « max », max, « rang max= », RangM

Fsi

Exercice VI : Ecrire un algorithme permettant de calculer la somme d'une suite d'entiers terminés par 0.

Solution :

Spécification :

- Données : Une suite d'entiers terminée par 0
- Résultat : La somme de cette suite

Solution en langage naturel : Tant que l'entier saisi est différent de 0, ajouter cet entier à la somme partielle et saisir le prochain entier.

Structure de données : Une première variable entière pour lire les entiers et un seconde pour calculer la somme partielle.

Algorithme :

Variables n, sommePartielle : numériques

sommePartielle ← 0

Lire (n)

Répéter

 sommePartielle ← sommePartielle + n

jusqu'à n = 0

Ecrire sommePartielle

Exercice VII : Ecrire un algorithme permettant de saisir une suite d'entiers terminée par l'entier 0 et de tester si cette suite contient deux entiers consécutifs égaux. Le programme s'arrête soit lorsque 2 entiers consécutifs égaux ont été saisis ou lorsqu'un entier nul a été saisi.

Solution :

1- Spécifications

- Données : Une suite d'entiers terminée par 0
- Résultats : Vrai s'il existe deux entiers consécutifs égaux ; faux sinon

2- Solution en langage naturel : Il faut comparer l'entier « courant » (celui que l'on vient de saisir) avec l'entier précédent tant qu'ils sont différents et que l'entier courant n'est pas nul.

3- Structure de données : Deux variables numériques n_courant et n_prec et une variable booléenne testant l'existence de deux entiers consécutifs égaux.

4- Algo :

Variables : ncour, nprec : numériques

 Lire ncour

 nprec ← ncour - 1

 tant que ncour ≠ 0 et nprec ≠ ncour faire

 nprec ← ncour

 Lire ncour

 Fin tant que

 Si ncour ≠ 0 alors écrire « La suite contient deux entiers consécutifs égaux »

 Sinon Ecrire « La suite ne contient pas deux entiers consécutifs égaux »

 Fsi