

Algorithmique

Chapitre X L'itération

1 - Faire de l'itération sans le savoir

Répéter

```
Lire a
c ← a * a
Ecrire c
Tant que a > 0
```

Les traitements sont
indépendants entre deux « tours
» du programme

Répéter

```
Lire a
S ← S + a
Tant que a > 0
Ecrire S
```

Les traitements sont dépendants
entre deux « tours » du
programme

1 - Faire de l'itération sans le savoir

- **Itération** : boucle dans laquelle « l'état » d'une ou plusieurs variables à un tour donné dépend de l'état au tour précédent
- Exemples :
 - $i \leftarrow i + 1$
 - Som \leftarrow Som + Val
- Les instructions de comptage et d'accumulation (voir chapitres précédents) sont des itérations particulières

2 - Un exemple simple d'itération

- Exemple : Calculer la somme des 100 premiers nombres entiers
 $1 + 2 + 3 + \dots + 99 + 100$

Solution 1

```
s ← 0
Répéter pour i = 1 à 100
    s ← s + i
Fin pour
```

Solution 2

```
s ← 0
i ← 1
Répéter
    s ← s + i
    i ← i + 1
Tant que i <= 100
```

Solution 2 bis

```
s ← 0
i ← 0
Répéter
    i ← i + 1
    s ← s + i
Tant que i > 100
```

2 - Un exemple simple d'itération

- Etude de la progression de l'état des deux variables i et s (pour la solution 2bis)

```

s ← 0
i ← 0
Répéter
    i ← i + 1
    s ← s + i
Tant que i > 100
    
```

Instant	Valeur de i	Valeur de s
Avant d'entrer dans la boucle	0	0
Après 1 tour de boucle	1	1
Après 2 tours de boucle	2	1 + 2
...
Après k tours de boucle	k	1 + 2 + ... + k
Après $k+1$ tours de boucle	$k+1$	1 + 2 + ... + k + ($k+1$)
...
Après 100 tours de boucle	100	1 + 2 + ... + 99 + 100

18/09/2005

Gilles.Gesquiere@up.univ-mrs.fr

5

2 - Un exemple simple d'itération

- Le passage d'un état au suivant est réalisé par les deux instructions :

- $i \leftarrow i + 1$
- $s \leftarrow s + i$

```

s ← 0
i ← 0
Répéter
    i ← i + 1
    s ← s + i
Tant que i > 100
    
```

i	s
0	0
1	1
...	...
k	1 + 2 + ... + k
$k + 1$	1 + 2 + ... + k + ($k+1$)

18/09/2005

Gilles.Gesquiere@up.univ-mrs.fr

6

3 - Un autre exemple : Calcul de factorielle

- Calculer la factorielle d'un nombre

- $n! = 1 * 2 * 3 * \dots * (n-1) * n$
- $1! = 1$
- $2! = 1 * 2 = 2$
- $3! = 1 * 2 * 3 = 6$
- ...
- $(k+1)! = k! * (k+1)$

- On utilise les instructions

- $i \leftarrow i + 1$
- $F \leftarrow F * i$

- Condition d'arrêt :

- Il faut que $i=100$ et $F=100!$
- $\Rightarrow i=100$

18/09/2005

Gilles.Gesquiere@up.univ-mrs.fr

7

3 - Un autre exemple : Calcul de factorielle

- Attention

- Aux instructions d'itérations
- Aux conditions d'arrêt

Solution 1

```

F ← 1
i ← 1
Répéter
    i ← i + 1
    F ← F * i
Tant que i > 100
    
```

Solution 2

```

F ← 1
i ← 1
Répéter
    F ← F * (i+1)
    i ← i + 1
Tant que i > 100
    
```



Solution 3 bis

```

F ← 1
Répéter pour i = 2 à 100
    F ← F * (i)
Fin pour
    
```



Solution 3

```

F ← 1
Répéter pour i = 1 à 99
    F ← F * (i+1)
Fin pour
    
```

18/09/2005

Gilles.Gesquiere@up.univ-mrs.fr

8

4 - La démarche itérative

- Cheminement d'un *état initial* vers un *état final* recherché
- Une boucle permet, par un même ensemble d'instruction, de progresser d'un état intermédiaire vers le suivant
- Pour « découvrir » une itération :
 1. Imaginer un état intermédiaire
 2. Voir comment progresser d'un état au suivant
 3. Définir le critère de fin de boucle
 4. Trouver comment commencer

18/09/2005

Gilles.Gesquiere@up.univ-mrs.fr

9

5 - Application : Recherche de maximum

- Exemple : on dispose de 50 valeurs, et l'on souhaite déterminer la plus grande de ces valeurs
- Recherche de la solution :
 1. Imaginer un état intermédiaire
 - Lorsque k valeurs ont été lues, le maximum est stocké dans la variable Max
 2. Voir comment progresser d'un état au suivant
 - A l'état suivant, le nouveau maximum sera stocké dans Max
 - Programme :

```
Lire Val
Si Val > Max alors Max ← Val Fsi
```

18/09/2005

Gilles.Gesquiere@up.univ-mrs.fr

10

5 - Application : Recherche de maximum

1. Définir le critère de fin de boucle
 - S'arrêter lorsque les 50 valeurs auront été examinées
 - ⇒ Utiliser un compteur du nombre de valeurs lues
 - Condition d'arrêt : $i = 50$
2. Trouver comment commencer
 - La valeur de Max est connue pour $k = 1$ (une seule valeur donc c'est la valeur maximale)

```
i ← 1
Lire Val
Max ← Val
```

18/09/2005

Gilles.Gesquiere@up.univ-mrs.fr

11

5 - Application : Recherche de maximum

```
Variables i, Max, Val : numériques
i ← 1
Lire Val
Max ← Val
Répéter
    i ← i + 1
    Lire Val
    Si val > Max alors Max ← Val Fsi
Tant que i <> 50
Ecrire « Maximum », Max
```

18/09/2005

Gilles.Gesquiere@up.univ-mrs.fr

12