

Découverte de l'informatique

TD3
14/10/16

Carlos Emiliano González Gallardo
carlos-emiliano.gonzalez-gallardo@univ-avignon.fr

1 Itérations déterministes

- **Exercice 1** Écrivez un algorithme utilisant une instruction itérative déterministe pour :

1. afficher à l'écran la factorielle d'un naturel n saisi en début de programme.

$n \leftarrow 6$

fact: 1*2*3*4*5*6

La factorielle de 6 est 720

1 Itérations déterministes

- **Exercice 1** Écrivez un algorithme utilisant une instruction itérative déterministe pour :

1. afficher à l'écran la factorielle d'un naturel n saisi en début de programme.

```
Entier n, i, fact
Afficher('Entrez une valeur :')
n←LireEntier()
1. fact←1
   pour i allant de 2 à n par pas de 1 faire
   | fact←fact * i
Afficher('La factorielle de ', n, ' est ', fact)
```

1 Itérations déterministes

- **Exercice 1** Écrivez un algorithme utilisant une instruction itérative déterministe pour :

2. afficher à l'écran tous les nombres entiers compris entre 1 et 100 qui sont des multiples d'un naturel n saisi en début de programme.

$n \leftarrow 10$

$a \bmod b$

10, 20, 30, 40, 50, 60, 70, 80, 90, 100

1 Itérations déterministes

- **Exercice 1** Écrivez un algorithme utilisant une instruction itérative déterministe pour :

2. afficher à l'écran tous les nombres entiers compris entre 1 et 100 qui sont des multiples d'un naturel n saisi en début de programme.

```
Entier n, i
```

```
Afficher('Entrez une valeur')
```

```
n ← LireEntier()
```

```
2. pour  $i$  allant de 2 à 100 par pas de 1 faire
```

```
    si  $i \bmod n = 0$  alors
```

```
        Afficher( $i$ , " ")
```

2 Itérations indéterministes

- **Exercice 2** Écrivez un algorithme en utilisation l'instruction itérative indéterministe qui vous semble la plus appropriée afin :

1. d'afficher la valeur u et le rang k du premier terme de la suite de Fibonacci dépassant une borne entière positive donnée p (saisie en début de l'algorithme)

index	0	1	2	3	4	5	6	7	8
suite	1	1	2	3	5	8	13	21	34

2 Itérations indéterministes

```
Entier precedent, rang, u, borne, temporaire
```

```
Afficher(“Entrez une borne maximale :”)
```

```
borne ← LireEntier()
```

```
u ← 1
```

```
rang ← 1
```

```
precedent ← 1
```

```
tant que  $u \leq$  borne faire
```

```
    tampon ← u
```

```
    u ← u + precedent
```

```
    precedent ← tampon
```

```
    rang ← rang + 1
```

```
Afficher(“Le premier terme de la suite de Fibonacci a dépasser “, borne, “ est celui  
de rang “, rang, “avec pour valeur “, u)
```

index	0	1	2	3	4	5	6	7	8
suite	1	1	2	3	5	8	13	21	34

borne ← 6

*Le premier terme de la suite de Fibonacci a dépasser **6** est celui de rang **5** avec la valeur **8***

2 Itérations indéterministes

- **Exercice 2** Écrivez un algorithme en utilisation l'instruction itérative indéterministe qui vous semble la plus appropriée afin :

2. de demander à l'utilisateur de deviner une valeur située entre 1 et 100. On considérera une valeur aléatoire obtenue en utilisation l'instruction `Aléatoire(100)` qui fournit un entier aléatoire entre 1 et 100 (on l'utilise donc de cette façon : `maVariableEntiere←Aléatoire(100)`). Un maximum de 10 tentatives est autorisé. A chaque tentative infructueuse, le programme indique si la valeur à trouver est supérieure ou inférieure à la valeur proposée.

2 Itérations indéterministes

```
Entier nombre, proposition, nbTentatives
nombre←Aléatoire(100)
nbTentatives←0
répéter
| Afficher('Entrez votre proposition : ')
| proposition←LireEntier()
| si proposition > nombre alors
| | Afficher('Trop grand')
| sinon
| | si proposition < nombre alors
| | | Afficher('Trop petit')
| nbTentatives←nbTentatives + 1
tant que proposition ≠ nombre et nbTentatives ≤ 10
si proposition = nombre alors
| Afficher('Bravo')
sinon
| Afficher('Vous avez perdu')
```

2 Itérations indéterministes

- **Exercice 2** Écrivez un algorithme en utilisation l'instruction itérative indéterministe qui vous semble la plus appropriée afin :

3. d'écrire le programme inverse du précédent où c'est au programme de deviner une valeur choisie par l'utilisateur. A chaque valeur proposée par le programme, l'utilisateur répond par le caractère '-', '=' ou '+' pour indiquer que sa valeur est respectivement inférieure, égale ou supérieure à la proposition.

2 Itérations indéterministes

Entier min, max, proposition, nbTentatives

Caractère retour

min←1

max←100

nbTentatives←0

Afficher(“Pensez à un nombre à trouver...”)

répéter

proposition←(min + max) div 2

Afficher(“Je propose “, proposition)

Afficher(“Dites-moi si c’est bon (=), trop petit (+) ou trop grand (-)”)

retour←LireCaractère()

si *retour* = ‘+’ **alors**

 | min←proposition

sinon

 | max←proposition

nbTentatives←nbTentatives + 1

tant que *retour* ≠ ‘=’

Afficher(“J’ai trouvé en “, nbTentatives, “ tentatives”)

carlos-emiliano.gonzalez-gallardo@univ-avignon.fr