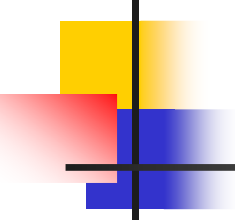


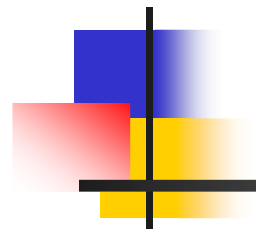


Algorithmique et programmation procédurale



Fonctionnel – procédural?

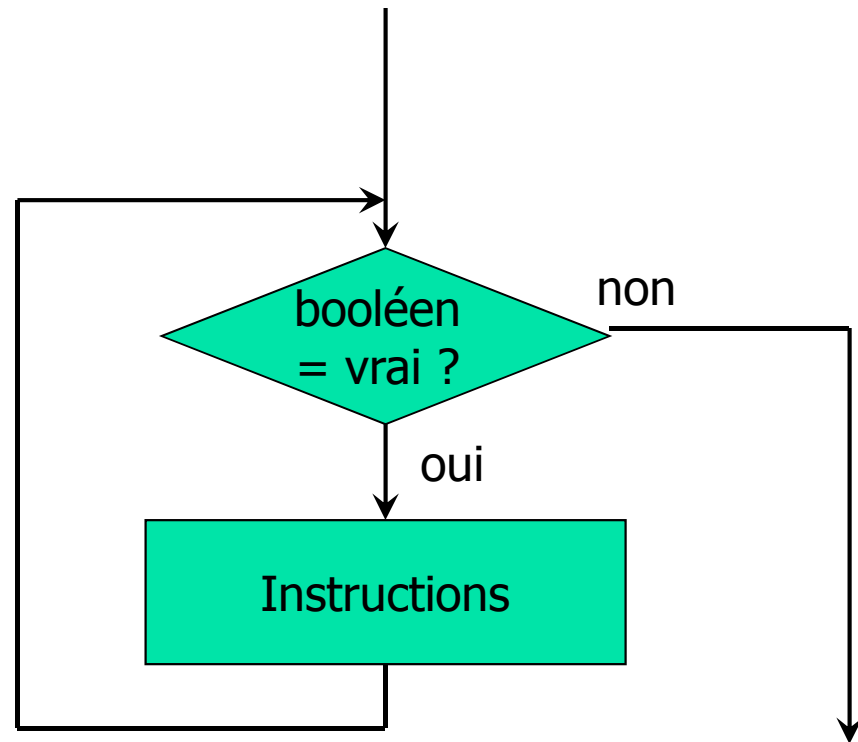
- Le **paradigme fonctionnel** considère qu'un programme est un ensemble de fonctions qui s'appellent entre elles afin de renvoyer un résultat final. La récursivité est au coeur de ce paradigme. Les langages les plus courants sont *Scheme* et *Caml*.
- Le **paradigme procédural** (ou **impératif**) est le plus ancien. Apparue avec *Fortran*, poursuivie avec *Pascal* et *C*, il met en avant l'appel de procédure pour effectuer un calcul séquentiel. La programmation procédurale considère qu'un texte de programme est une suite d'instructions pilotant un ordinateur, au contraire de la programmation fonctionnelle, où la notion d'instruction n'existe pas plus qu'en maths. L'itération est le mécanisme central pour effectuer un calcul répétitif.

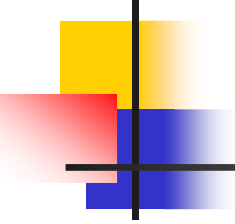


Cours 1: Boucles itératives

Boucle Tant que

...
Tantque booléen
Instructions
FinTantque
...





Exemple : contrôle de saisie

Programme BoucleTantQue

Variables Rep : Caractère

Début

 Ecrire "Voulez-vous un café ? (O/N)"

 Lire Rep

 Tantque Rep ≠ "O" et Rep ≠ "N"

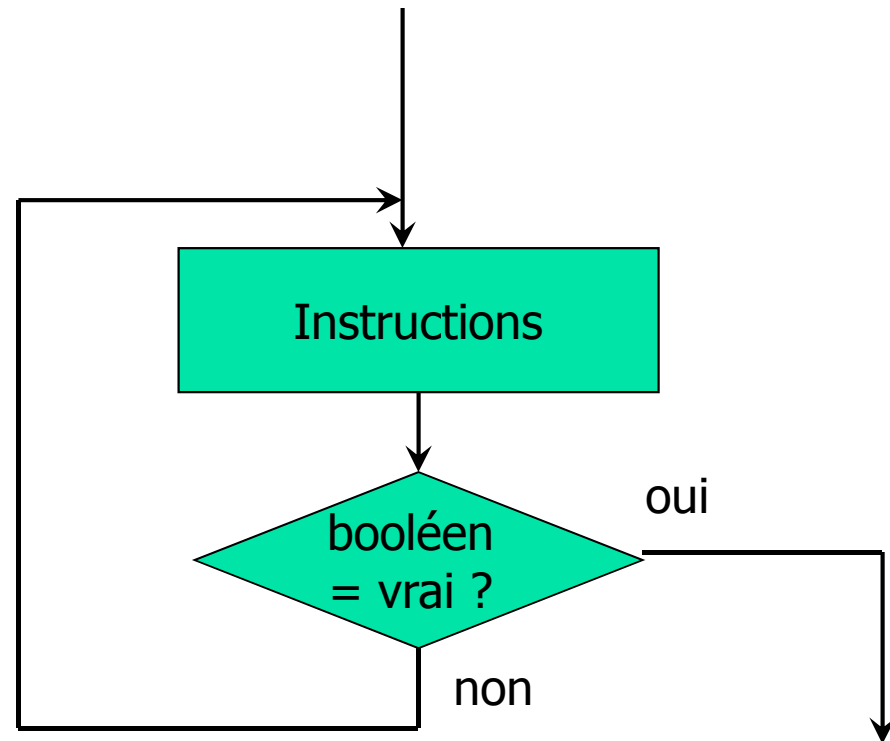
 Lire Rep

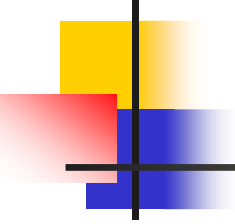
 FinTantque

Fin

Boucle Jusqu'à ce que

...
Répète
Instructions
Jusqu'à ce que booléen
...





Exemple : contrôle de saisie

Programme BoucleRepete

Variables Rep : Caractère

Début

Ecrire "Voulez-vous un café ? (O/N)"

Répète

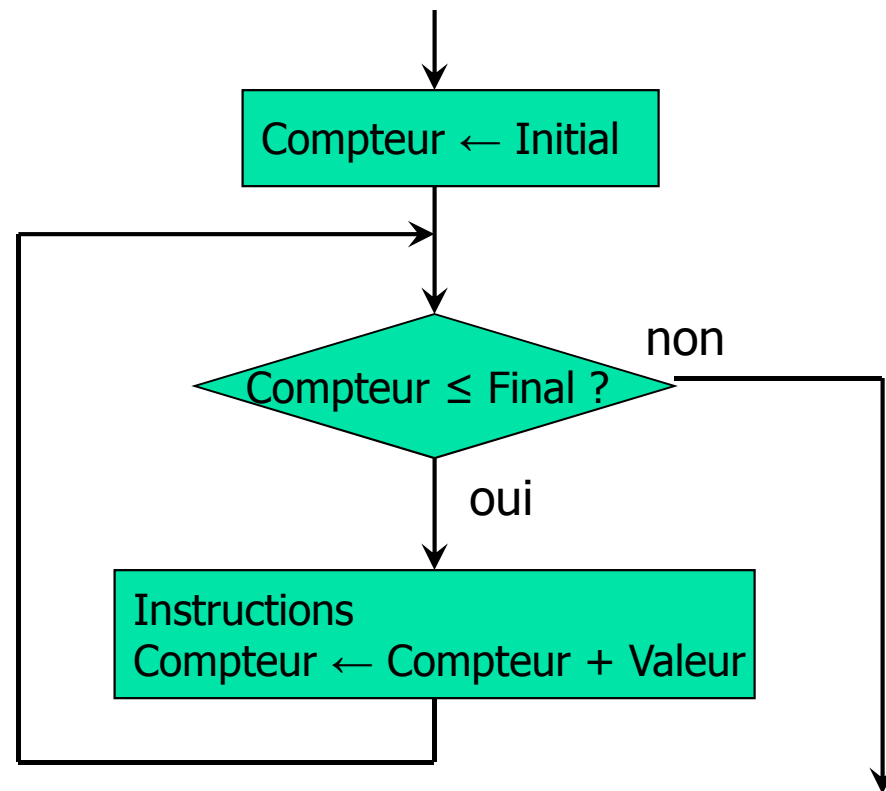
Lire Rep

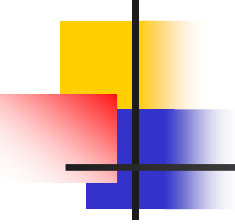
Jusqu'à ce que Rep = "O" ou Rep = "N"

Fin

Boucle Pour

...
Pour Compteur $\leftarrow$ Initial à Final **Pas** Valeur
Instructions
FinPour
...





Exemple

Programme BouclePour

Variables N, i, Som : **Entier**

Début

Ecrire "Entrez un nombre : "

Lire N

 Som $\leftarrow$ 0

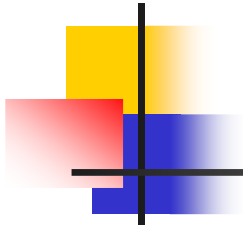
Pour i $\leftarrow$ 1 à N

 Som $\leftarrow$ Som + i

FinPour

Ecrire "La somme est : ", Som

Fin



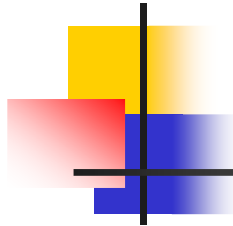
Exercice

- Ecrire une fonction qui renvoie, pour un entier N donné, le produit

$$1*3*5* \dots *(2N+1)$$

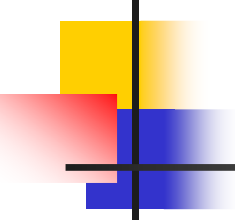
en utilisant:

- Une récursion
- Une boucle « Tantque »
- Une boucle « Répète-Jusqu'à ce que »
- Une boucle « Pour »



Complexité?

- Comme dans le cas de la récursivité, il est important de considérer la complexité d'une boucle



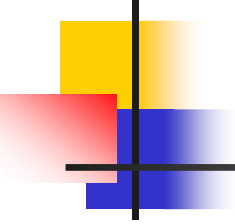
Exercice

Combien de multiplications sont effectuées dans chacune de ces séquences (x^{2^n}) ?

```
X2N ← 1
Pour i ← 1 à 2*N
    X2N ← X2N*X
FinPour
```

```
X2N ← 1
X ← X*X
Pour i ← 1 à N
    X2N ← X2N*X
FinPour
```

```
XN ← 1
Pour i ← 1 à N
    XN ← XN*X
FinPour
X2N ← XN*XN
```



Exercice

Quelle est la complexité de cet algorithme en nombre d'additions et de divisions ?

resultat $\leftarrow$ 0

Pour $i \leftarrow 1$ à n (1)

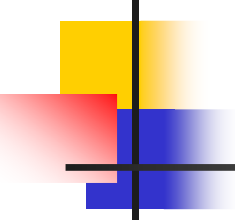
 Pour $j \leftarrow i$ à $i + 2$ (2)

 resultat $\leftarrow$ resultat + $1/(i + j)$ (3)

 FinPour

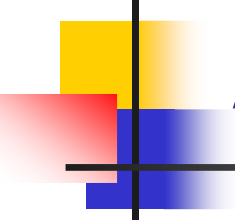
FinPour

Ecrire resultat



Exemple: le compteur « LE »

- Calculer le nombre d'occurrences de la chaîne de caractères "LE" dans une chaîne de caractères dont la fin est un symbole spécial noté *
- Opérations :
 - tests (comparaisons)
 - opérations logiques
 - opérations arithmétiques
 - affectations
 - lecture / écriture



Algorithme 1

NLE $\leftarrow$ 0

IlyaunL $\leftarrow$ faux

Car $\leftarrow$ lire-premier-caractère

Tantque Car $\neq$ "*" **FinTantque**

Selon Car

cas "L" : IyaunL $\leftarrow$ vrai

cas "E" : **Si** IyaunL **Alors** NLE $\leftarrow$ NLE +1 **FinSi**

IyaunL $\leftarrow$ faux

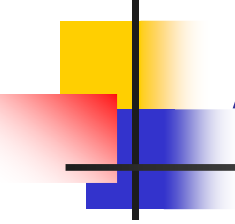
autres : IyaunL $\leftarrow$ faux

FinSelon

Car $\leftarrow$ lire-caractère-suivant

FinTantque

Ecrire NLE



Algorithme 2

NLE $\leftarrow$ 0

Carprec $\leftarrow$ " "

Car $\leftarrow$ lire-premier-caractère

Tantque Car $\neq$ "*" **Alors**

Si (Carprec = "L") **et** (Car = "E") **Alors**

 NLE $\leftarrow$ NLE + 1

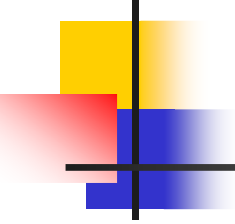
FinSi

 Carprec $\leftarrow$ Car

 Car $\leftarrow$ lire-caractère-suivant

FinTantque

Ecrire NLE



Comparaisons

	Algorithme 1	Algorithme 2
Tests	$N + 1 + NE + N$	$N + 1 + 2N$
Op. logiques	0	N
Op. arithmétiques	NLE	NLE
Affectations	$2 + NLE + N$	$2 + NLE + N$
Lecture/écriture	$N + 2$	$N + 2$

- N : nombre de caractères de la chaîne (N+1 si l'on compte le caractère *)
- NE : nombre de E
- NLE : nombre recherché de LE