Cours 2

M.Adam – JF.Kamp – S.Letellier – F.Pouit

30 juin 2016

Table des matières

1 La boucle

1.1 Calcul du PGCD

Le PGCD, Plus Grand Commun Diviseur, de 50 et 125.

1.1.1 Algorithme de calcul du PGCD

Le PGCD de deux nombres est obtenu en soustrayant le plus petit des deux nombres au plus grand jusqu'à ce que les deux soient égaux.

50	125

Il nous manque une structure de contrôle pour permettre de réaliser cet algorithme en TestAlgo : la boucle.

1.2 La boucle ou l'itération

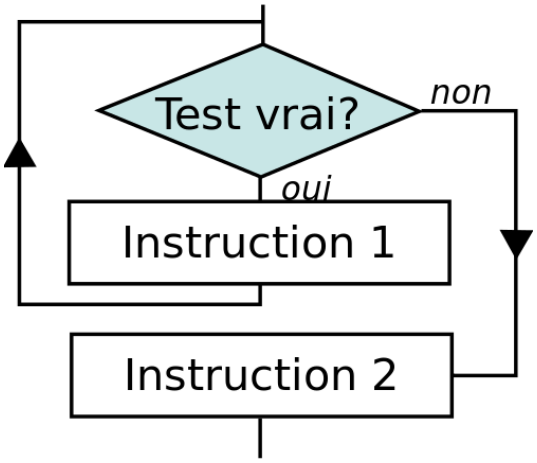
Syntaxe :

```
tantque (condition)
  instructions
fintantque
```

1.2.1 Interprétation d'une boucle

Répéter les `instructions` tant que la `condition`, `Test`, est vraie.

```
tantque (Test)
    instruction 1;
fintantque
instruction 2;
```



1.2.2 Premier exemple

Le programme soustrait 1 à la variable `i` jusqu'à ce qu'elle soit égale à 0.

```
##
# décrémente i jusqu'à 0
# @author M.Adam
##
algo Ex1
principal
var
    entier i;
debut
    saisir("Valeur de i",@i);
    tantque (i <> 0)
        afficherln("Valeur de i : "+i);
        i:=i-1;
    fintantque
    afficherln("Valeur de i : "+i);
fin
```

```
TestAlgo - Interprétation engagée.
3
Valeur de i : 3
Valeur de i : 2
Valeur de i : 1
Valeur de i : 0

TestAlgo - Fin de l'interprétation.
```

Instructions	i	écran et clavier
`saisir("Valeur de i",@i);`	3	3
`tantque (i <> 0)`	3	
`i:=i-1;`		

1.3 Les différentes parties d'une boucle

- L'initialisation

```
saisir("Valeur de i",@i);
```

- La condition de continuation :

```
i <> 0
```

- La condition d'arrêt :

`non i <> 0` qui est équivalent à `i == 0`

- Le corps de boucle :

```
afficherln("Valeur de i : "+i);
i:=i-1;
```

- La terminaison :

```
afficherln("Valeur de i : "+i);
```

1.4 Autres formes de boucles

Il existe d'autres boucles :

- `repeter ... jusqua`
- `pour`

Ce ne sont que d'autres formes de la boucle `tantque`.

1.5 Les restrictions appliquées sur cette séquence

- Une seule condition de continuation
- Pas de boucles imbriquées

2 Les limites des boucles

2.1 Les risques d'erreurs

- La boucle infinie,

- l'erreur de calcul.

2.2 Boucle infinie

- Il faut que le corps de la boucle rende la condition de continuation fausse,
- mais ce n'est pas évident !

2.2.1 Boucle ou pas ?

```
##
# décrémente i jusqu'à 0
# @author M.Adam
##
algo Ex1
principal
var
    entier i;
debut
    saisir("Valeur de i",@i);
    tantque (i <> 0)
        afficherln("Valeur de i : "+i);
        i:=i-1;
    fintantque
    afficherln("Valeur de i : "+i);
fin
```

Instructions	i	écran et clavier
`saisir("Valeur de i",@i);`		
`tantque (i <> 0)`		
`i:=i-1;`		

2.2.2 Boucle ou pas ?

```
tantque (i <> j)
  i := i+1;
  j := j-1;
fintantque
```

2.2.3 Boucle ou pas ?

```
tantque (i <> 0)
  si (i < 0) alors
      i := i+1;
  sinon
      i := i-1;
  finsi
fintantque
```

2.2.4 Boucle ou pas ?

```
saisir("rep : ",@rep);
tantque (rep <> 'o' ou rep <>'n')
    saisir("rep : ",@rep);
fintantque
```

La condition de sortie est :

Pas facile de gérer les conditions de continuation avec plusieurs parties disjonctives ou conjonctives !

2.3 L'erreur de calcul

Le résultat rendu pas la boucle n'est pas celui attendu.

2.3.1 Correct ou pas ?

```
##
# Calcul de a^n
# @author M.Adam
##
algo Exposant
principal
var
    reel a, exp;
    entier  n, i;
debut
    afficherln("Calcul de a^n");
    saisir("Valeur de a",@a);
    saisir("Valeur de n",@n);
    exp := 1;
    i := 1;
    tantque (i <> n)
        exp := exp*a;
        i:=i+1;
```

```
    fintantque
    afficherln("Valeur de a^n : "+exp);
fin
```

2.3.2 Correct ou pas ?

Nous allons résoudre par essais successifs l'équation $x^3 + x^2 = 100$.

- La fonction f est définie par $f(x) = x^3 + x^2$ sur $[0\,;\,10]$.
- Cette fonction est croissante sur $[0\,;\,10]$ et on peut calculer que : $f(4) = 80$ et $f(5) = 150$.
- Par conséquent on peut trouver une solution approximative de l'équation $x^3 + x^2 = 100$ sur $[4\,;\,5]$. Nous allons donc procéder de la manière suivante :
  - calculer $f(4)$
  - puis calculer $f(4.01)\,;\,f(4.02)\,;\,f(4.03)$ et ainsi de suite, et de faire continuer l'algorithme tant que l'image de chaque nombre est inférieure à 100.

```
##
# Calcul de x*x*x+x*x=100 par approximation
# @author M.Adam
##
algo Approximation
principal
var
    reel x, f;
debut
    x := 4;
    f := x*x*x+x*x;

    tantque (f < 100)
        x := x + 0.01;
        f := x*x*x+x*x;
    fintantque
    afficherln("f("+x+") = "+f);
fin
```

3 Conclusion

3.1 En résumé

- La boucle `tantque`
- Les risques d'erreurs

3.2 A venir

- La construction des boucles
- La structure de tableau