



TD Boucles

Résumé

Ces exercices ont pour but de vérifier que vous avez fixé les structures alternatives qui permettent de conditionner des parties d'algorithmes, de code.

1	Alternatives	2
1.1	Compréhension d'algorithme	2
1.2	Compréhension de codes Java	3
1.3	À vous de jouer...	7



1 Les boucles

1.1 Compréhension d'algorithme

Pour ces exercices, nous vous demandons de comprendre des algorithmes donnés.

Compréhension

Que vont-ils afficher ?

```
— module boucle1 ()
    x : entier
    x ← 0
    tant que x < 12 faire
        x ← x+2
    fin tant que
    afficher x
fin module

—
— module boucle2 ()
    ok : booléen
    x : entier
    ok ← vrai
    x ← 5
    tant que ok faire
        x ← x+7
        ok ← x MOD 11 ≠ 0
    fin tant que
    afficher x
fin module

—
— module boucle3 ()
    ok : booléen
    cpt, x : entiers
    x ← 10
    cpt ← 0
    ok ← vrai
    tant que ok ET cpt < 3 faire
        si x MOD 2 = 0 alors
            x ← x+1
            ok ← x < 20
        sinon
            x ← x+3
```

```

        cpt ← cpt + 1
    fin si
    fin tant que
    afficher x
fin module

— module boucle4 ()
    pair, grand : booléens
    p, x : entiers
    x ← 1
    p ← 1
    faire
        p ← 2*p
        x ← x+p
        pair ← x MOD 2 = 0
        grand ← x > 15
    jusqu'à ce que pair OU grand
    afficher x
fin module

— module boucle5 ()
    i, x : entiers
    ok : booléen
    x ← 3
    ok ← vrai
    pour i de 1 à 5 faire
        x ← x+i
        ok ← ok ET (x MOD 2 = 0)
    fin pour
    si ok alors
        afficher x
    sinon
        afficher 2 * x
    fin si
fin module

— module boucle6 ()
    i, j, fin : entiers
    pour i de 1 à 3 faire
        fin ← 6 * i - 11
        pour j de 1 à fin par 3 faire
            afficher 10 * i + j
        fin pour
    fin pour

```

```
fin module
```

1.2 Compréhension de codes Java

Instructions répétitives

Quelles instructions répétitives sont correctes parmi les suivantes ? Expliquez pourquoi les autres ne le sont pas.

- ☐ proposition 1

```
While ( condition ) {  
    // instructions  
}
```

- ☐ proposition 2

```
do while ( condition ) {  
    // instructions  
}
```

- ☐ proposition 3

```
while ( true ) {  
    // instructions  
}
```

- ☐ proposition 4

```
while ( true ) do {  
    // instructions  
}
```

- ☐ proposition 5

```
FOR ( int i=0; i<=10; i=i+2 ) DO {  
    // instructions  
}
```

- ☐ proposition 6

```
for ( int i=0; i<=10; i=i+2 ) {  
    // instructions  
}
```

- ☐ proposition 7

```
for ( int i=0; i<=10; i=i+2 ) do {  
    // instructions  
}
```

- ☐ proposition 8

```
for ( int i=9; i>=0; i=i-2 ) {  
    // instructions  
}
```

Activité 'remplir les blancs'

Quel opérateur de comparaison Java représente la relation suivante ?

1. "est égal à" ? ____
2. "est différent de" ? ____

Quel opérateur booléen Java représente l'opérateur logique suivant ?

1. le ET : ____
2. le OU : ____
3. le NON : ____

Expérience

Indiquez l'affichage obtenu par ce code.

Compréhension

Que vont-ils afficher ?

```
public class Boucles {  
    public static void main ( String[] args ) {  
        int facteur;  
        final int VALEUR = 3;  
  
        for (facteur = 1 ; facteur <= 10 ; facteur++){  
            System.out.print(facteur*VALEUR+" ");  
        }  
        System.out.println();  
    }  
}
```

Exercice Tant que

Écrivez en Java l'algorithme suivant.

MODULE Test

```
nb, produit : Entier  
produit ← 1  
  
LIRE nb  
TANT QUE nb ≠ 0 FAIRE
```

```

        produit ← produit * nb
        LIRE nb
    FIN TANT QUE
    AFFICHER produit

FIN MODULE

```

Exercice Pour

Écrivez en Java l'algorithme suivant.

```

MODULE Test

    nb: Entier
    i : Entier

    LIRE nb
    POUR i DE 1 A nb FAIRE
        AFFICHER i
    FIN POUR

FIN MODULE

```

1.3 À vous de jouer...

À la pompe

À la pompe à essence, le prix du carburant dépend du type de carburant. Reprenons ci-dessous le prix au litre pratiqué par ESI-Pompe.

- Super 95 : 1,429 €/L
- Super 98 : 1,604 €/L
- Diesel : 1,249 €/L
- LPG : 0,558 €/L

Mettez en évidence les variables «**données** », les variables «**résultats** » et les variables de travail ;

Écrivez un module qui lit le type de carburant et la quantité désirée (considérée positive) et qui affiche le prix à payer.

Exemple : pour du Super 95 et pour une quantité de 30,46 litres le module retourne le prix de 43,52 €.

Écrivez le code java correspondant.

Métro

L'horaire du métro bruxellois varie selon le jour et l'heure. La fréquence des passages est de 5 minutes en semaine de 6h à 10h et de 15h à 19h. De 7 minutes en semaine de 10h à 15h et le samedi de 12h à 19h. De 10 min après 19h en semaine et le samedi et toute la journée de dimanche.

Mettez en évidence les variables «**données** », les variables «**résultats** » et les variables de travail ;

Écrivez un module qui reçoit un nom de jour et une heure et qui retourne la fréquence pour ce moment.

Exemple : jour = «lundi » ; heure = 16 fréquence = 5

Écrivez le code java correspondant.

La conjecture de Goldbach

La conjecture de Goldbach est une assertion mathématique non démontrée qui s'énonce comme suit :

Tout nombre entier pair supérieur à 3 peut s'écrire comme la somme de deux nombres premiers.

Écrivez un module `isPremier` qui reçoit un nombre entier `n` et qui retourne vrai si ce nombre est premier et faux sinon.

Écrivez un module `goldbach` qui reçoit en paramètre un nombre entier pair `p` supérieur à 3 et qui retourne vrai s'il est la somme de 2 nombres premiers et faux sinon. Si le `p` reçu n'est supérieur à 3, votre programme générera une erreur.

Mettez en évidence les variables «**données** », les variables «**résultats** » et les variables de travail ;

Écrivez un module qui vérifie si trois entiers donnés constituent un triple Pythagoricien.

Écrivez le code java correspondant.

Pour plus d'exercices, révisez ici (www.heb.be/esi/InitBoucle/fr/../../TDA1t/fr/html/unit_Exercices.html)