

Algorithmes Séance 4 - Correction

Problèmes

Compteur de 1 à 3

Ecrire un programme qui compte de 1 à 3

```

Algorithme Ex1
Début
  Ecrire(1)
  Ecrire(2)
  Ecrire(3)
Fin

```

Compteur de 1 à n

Ecrire un programme qui saisit un nombre n, puis compte de 1 à n.

```

Algorithme Ex2
Début
  Ecrire("Donner un entier n ? "); Lire(n)
  Pour i de 1 à n faire
    Ecrire(i)
  Fin Pour
Fin

```

FizzBuzz

Écrire l'algorithme d'un programme qui affiche les nombres de 1 à 100. Mais pour les multiples de trois, afficher « Fizz » au lieu du nombre et pour les multiples de cinq, afficher « Buzz ». Pour les nombres qui sont des multiples de trois et de cinq, afficher « FizzBuzz ».

1^{ère} Méthode

```

Algorithme Ex3
Début
  Pour i de 1 à 100 faire
    Si n mod 15 = 0 Alors Ecrire("FizzBuzz") Fin Si
    Sinon Si n mod 3 = 0 Alors Ecrire("Fizz") Fin Si
    Sinon Si n mod 5 = 0 Alors Ecrire("Buzz") Fin Si
    Sinon Ecrire(i)
  Fin Si
  Fin Pour
Fin

```

2^{ème} Méthode

```

Algorithme Ex3
Début
  Pour i de 1 à 100 faire
    res ← ""
    Si n mod 3 = 0 Alors res ← "Fizz" Fin Si
    Si n mod 5 = 0 Alors res ← res + "Buzz" Fin Si
    Si res = "" Alors res ← convch(i) Fin Si
    Ecrire(res)
  Fin Pour
Fin

```

Nombres Palintiples

Un nombre **palintiple** est un nombre multiple de son retourné.

Exemples : Pour les nombres de quatre chiffres les palintiples sont : $1089 \times 9 = 9801$ et $2178 \times 4 = 8712$

Ecrire l'algorithme d'un programme qui cherche tous les nombres palintiples de 5 chiffres.

1^{ère} Méthode

Algorithme Ex4

Début

Pour i de 10000 à 99999 faire

u ← i mod 10

d ← (i div 10) mod 10

c ← (i div 100) mod 10

m ← (i div 1000) mod 10

dm ← i div 10000

ii ← u * 10000 + d * 1000 + c * 100 + m * 10 + dm

Si (i ≠ ii) et (ii ≥ 10000) et (i mod ii = 0) Alors

Ecrire(i, "=", ii, "x", i div ii)

Fin Si

Fin Pour

Fin

2^{ème} Méthode

Algorithme Ex4

Début

Pour i de 10000 à 99999 faire

chi ← convch(i)

ii ← valeur(chi[4]+chi[3]+chi[2]+chi[1]+chi[0])

Si (i ≠ ii) et (ii ≥ 10000) et (i mod ii = 0) Alors

Ecrire(i, "=", ii, "x", i div ii)

Fin Si

Fin Pour

Fin

3^{ème} Méthode

Algorithme Ex4

Début

Pour dm de 1 à 9 faire

Pour m de 0 à 9 faire

Pour c de 0 à 9 faire

Pour d de 0 à 9 faire

Pour u de 0 à 9 faire

i ← dm * 10000 + m * 1000 + c * 100 + d * 10 + u

ii ← u * 10000 + d * 1000 + c * 100 + m * 10 + dm

Si (i ≠ ii) et (ii ≥ 10000) et (i mod ii = 0) Alors

Ecrire(i, "=", ii, "x", i div ii)

Fin Si

Fin Pour

Fin Pour

Fin Pour

Fin Pour

Fin Pour

Fin

Nombre Parfait

Les nombres parfaits sont des entiers égaux à la somme de leurs diviseurs. Ainsi, 6 se divise par 2, 3 et 1. En additionnant 2, 3 et 1, on arrive à 6 ! Même chose pour 28, somme de $1 + 2 + 4 + 7 + 14$.

Ecrire l'algorithme d'un programme qui saisit un nombre puis affiche s'il est parfait ou non.

Exemples

Donner un entier ? 6
Somme des diviseurs : 6

6 est un nombre parfait!

```
Algorithme Ex5
Début
  Ecrire("Donner un nombre ? "); Lire(n)
  s ← 0
  Pour i de 1 à n div 2 faire
    Si n mod i = 0 Alors
      s ← s + 1
  Fin Pour
  Si n = s Alors
    Ecrire(n, "est un nombre parfait")
  Sinon
    Ecrire(n, "n'est pas un nombre parfait")
  Fin Si
Fin
```

Nombre de mots dans une phrase

La phrase suivante :

Ikram est allée au supermarché.

est composée de 5 mots.

Ecrire l'algorithme d'un programme qui calcule le nombre de mots dans une phrase saisie par l'utilisateur.

```
Algorithme Ex6
Début
  Ecrire("Donner une phrase ? "); Lire(ph)
  nbm ← 1
  Pour i de 0 à Long(ph)-1 faire
    Si ph[i] = " " Alors
      nbm ← nbm + 1
  Fin Pour
  Ecrire("La phrase contient", nbm, "mots")
Fin
```