Travaux dirigés n° 2

Boucles

Exercice 1 (Premières boucles)

On considère les algorithmes suivants :

```
Algorithme Algorithme 1
Déclarations
      Constantes
            n = 5
      Variables
            p, i, resultat : entier
Début
{1}   p ← 1
{2}   i ← n
{3}   TantQue i ≠ 0 Faire
{3.1}       p ← p * n
{3.2}       i ← i - 1
      FinTantQue
{4}   resultat ← p
Fin
```

```
Algorithme Algorithme 2
Déclarations
      Constantes
            n = 5, x = 2
      Variables
            p, i, resultat : entier
Début
{1}   p ← 1
{2}   Pour i allant de 1 à n Faire
{2.1}       p ← p * x
      FinPour
{3}   resultat ← p
Fin
```

1°) Dressez le tableau d'exécution de ces algorithmes.

2°) De façon générale, déroulez ces algorithmes avec $n = k \geq 1$.

3°) Que se passe-t-il si $k < 0$? Si $k = 0$? Si $k = 1$?

4°) Écrivez un algorithme équivalent à l'algorithme 1 en utilisant une boucle "pour".

5°) Écrivez un algorithme équivalent à l'algorithme 2 en utilisant une boucle "tant que".

Exercice 2 (PGCD d'Euclide)

Écrivez l'algorithme d'Euclide du calcul du PGCD de deux entiers strictement positifs. Vous utiliserez trois variables a, b et r où a et b sont les deux entiers et r est le reste de la division de a par b.

Indications : cet algorithme consiste à calculer r qui est le reste de la division de a par b. On remplace ensuite a par b puis b par r. On recommence l'opération jusqu'à ce que r soit égal à 0. Le PGCD est alors égal à b.

Exercice 3 (Représentation binaire d'un entier naturel)
On désire obtenir la représentation binaire d'un entier naturel.

1°) Rappelez la méthode sur un exemple.

2°) On notera a_i le i-ième terme de la représentation binaire (c'est-à-dire le coefficient de 2^i).

a) On considère que le nombre de bits est fixé (8, 16, ...) : proposez un algorithme de calcul des a_i (vous les afficherez au fur et à mesure).

b) Proposez un algorithme qui permet de calculer uniquement les bits nécessaires (*i.e.* les 0 non significatifs ne sont pas affichés).

Exercice 4 (Puissance)
1°) Proposez un algorithme de calcul de x^n (n positif ou négatif).

2°) Combien de multiplications demande l'exécution de cet algorithme ?

3°) Proposez un algorithme calculant x^{10} en 4 multiplications.

Exercice 5 (Syracuse)
On considère la suite de Syracuse, définie par la donnée de $S_0 \in \mathbb{N}^*$ et par la relation de récurrence : $S_{n+1} = S_n/2$ si S_n est pair ; $S_{n+1} = 3S_n + 1$ si S_n est impair.

1°) Écrivez un algorithme permettant de saisir S_0 et d'afficher les termes jusqu'à S_{10} (on se limite aux seules variables entières S et n).

2°) Transformez l'algorithme de manière à afficher les termes dix par dix tant que l'utilisateur veut continuer.

3°) Faites afficher un terme, puis deux, puis trois, etc ..., tant que l'utilisateur veut continuer.

4°) Faites afficher tous les termes jusqu'à rencontrer la valeur 1 pour la première fois ; on affichera alors des statistiques sur les termes calculés jusqu'alors : nombre de termes, moyenne, maximum, minimum, ...

Exercice 6 (Au carré)
La somme des n premiers nombres impairs est égale au carré de n.

Exemple : pour $n = 3$, $S = 1 + 3 + 5 = 9 = 3^2$

1°) Écrivez un algorithme calculant n^2, $n > 0$ étant saisi au clavier, en utilisant cette propriété.

2°) Que se passe-t-il si $n = 0$? Et si $n < 0$? Modifiez votre algorithme pour gérer ces cas.