Initiation à l'algorithmique

TD 2

Exercice 2.1

On considère la fonction suivante :

```
fonction test (ref n:entier;val i:entier):vide;
debut
   tant que i > 0 faire
     n = n + i;
     i = i-1;
   fintantque
fin
```

Que valent n et i après l'appel de la fonction `test` dans les cas suivants :

1. ```
   var n,i:entier
   test(3,5)
   ```
2. ```
   var n,i:entier
   n = 4
   test(n,5)
   ```
3. ```
   var n,i:entier
   n = 4
   i = 5
   test(n,i);
   ```

Exercice 2.2

Ecrire les fonctions `puissance`, `Modulo`, `QuotientEntier`, `PGCD` à partir des algorithmes décrit dans le TD1.

Exercice 2.3

Multiplication alexandrine
(ou Multiplication à la russe)

Pour multiplier deux entiers X et Y, on détermine la suite :
- $(x_0, y_0) = (X, Y)$
- $(x_{n+1}, y_{n+1}) = (2x_n, y_n/2)$ (où le quotient est un quotient entier)

Le produit XY est égal à la somme des x_i pour lesquels y_i est impair.

a- Appliquer l'algorithme au produit 7×9.
b- Justifier l'algorithme ; expliquer son intérêt dans un calculateur.
c- Ecrire une fonction itérative `Produit` appliquant cet algorithme. Quel est le paramètre pertinent pour évaluer le temps d'exécution ?
e- Déduire des questions précédentes un algorithme permettant d'obtenir une fonction retournant x^y.
   ```

Exercice 2.4

Quelle est la plus petite valeur strictement positive de n pour laquelle un algorithme dont le temps d'exécution est $100n^2$ s'exécute plus vite qu'un algorithme pour lequel le temps d'exécution est 2^n sur la même machine ?
Soit $n_1(t)$ (resp. $n_2(t)$) la valeur maximum de n pour laquelle on peut exécuter un certain algorithme réalisant $100n^2$ (resp. 2^n) opérations en un temps t. De combien augmente l'image de la fonction n_1 (resp. n_2) lorsque l'on passe de t à $2t$?

Exercice 2.5

Écrire une fonction qui teste si un entier est premier ou non.

Exercice 2.6

Écrire une fonction utilisant les précédentes permettant d'afficher le énième nombre premier.