
Exercices : Exercices de base de données

Exercice 1 Ecrire un algorithme qui calcule $n!$ à l'aide d'une boucle while. On prouvera la terminaison et la correction de celui-ci

Exercice 2 (Division entière) Rappelons tout d'abord, la définition de la division entière de deux entiers naturels a et b :

$$\forall a \geq 0, b > 0, a = q.b + r, 0 \leq r < b$$

Pour calculer la division entière, nous allons procéder par soustractions successives de la valeur b de a jusqu'à ce que le résultat de la soustraction soit inférieur à b .

On considère l'algorithme :

```
def divisionEntier(a,b)
    q=1
    r=a
    while r>=b:
        q=q+1
        r=r-b
    return (q,r)
```

1. Exhiber un invariant de boucle.
2. Montrer que $a=q.b+r$ est un invariant de boucle.

Exercice 3 On considère l'algorithme suivant où toutes les variables contiennent des entiers et $n > 1$.

```
def racine_carre(n):
    inf=1
    sup=n
    while inf < sup-1:
        mid=(inf + sup) // 2
        if mid*mid <= n:
            inf=mid
        else :
            sup=mid
    return inf
```

1. Montrer que l'invariant $inf^2 \leq n < sup^2$ est vrai au départ de la boucle et conservé ensuite.
2. En déduire que si le programme termine; il retourne $\lfloor \sqrt{n} \rfloor$.
3. Montrer que la différence $sup - inf$ diminue strictement à chaque étape de boucle. En déduire que le programme termine.

Exercice 4 Le produit de deux entiers x et y consiste à sommer y fois la valeur x . Toutefois, on peut améliorer cet algorithme rudimentaire en multipliant x par deux et en divisant y par deux chaque fois que sa valeur est paire. Les opérations de multiplication et de division par deux sont des opérations très efficaces en informatique puisqu'elles consistent à décaler de un bit vers la gauche ou vers la droite. Le programme correspondant à cet algorithme est le suivant :

```
def produit(x,y)
    a=x
    b=y
    prod=0
    while b>0:
        while b%2==0:
            b=b//2
            a=2*a
        prod=prod+a
        b=b-1
    return prod
```

1. Déterminer des variant de boucle pour chacune.
2. Montrer ab constant est un invariant de boucle de 2ème while.
3. Montrer que $prod+ab=xy$ est un invariant du deuxième while.

Exercice 5 1. (a) écrire une fonction qui prend en paramètre un nombre a et un entier positif ou nul n et qui retourne a^n .
 (b) à l'aide d'un invariant de boucle montrer que l'algorithme est correct.
 (c) Donner le variant de votre boucle.

2. On considère maintenant la fonction suivante

```
def puissance2(a,n):
    n2 = n
    a2i = a
    res = 1
    while n2 > 0 :
        if n2%2 == 1:
            res = res*a2i
        n2 = n2 // 2
        a2i = a2i*a2i
    return res
```

Montrer que à la fin de chaque étape de boucle, on a $a^n = res \times a2i^{n2}$.

Exercice 6 On considère l'algorithme TriLent qui trie (en ordre croissant) un tableau T , indicé de 0 à n .

```
def TriLent(T,n):  
    i=0  
    while i<n:  
        if T[i+1]>=T[i]:  
            i=i+1  
        else:  
            t=T[i]  
            T[i]=T[i+1]  
            T[i+1]=t  
            i=0
```

1. Donner un convergent.
2. Montrer que « la partie du tableau de $T[1]$ à $T[i]$ est triée en ordre croissant » est un invariant de boucle.