

Informatique PCSI

TP 10a : analyse d'un programme

Exercice 1

Dans la fonction suivante, les valeurs des variables a et b sont des entiers naturels :

```
def f(a, b):
    s, k = a, b
    while k > 0:
        s = s + 1
        k = k - 1
    return s
```

Quelle affirmation est fautive ?

1. La propriété « $s+k = a+b$ » est un invariant de la boucle `while`.
2. La valeur finale de k est 1.
3. La propriété « $k \geq 0$ » est un invariant de la boucle `while`.
4. Le résultat renvoyé est égal à la somme $a+b$.

Exercice 2

On considère la fonction `dichotomie` définie ci-dessous. Tester cette fonction avec des exemples simples afin de déterminer un bug et le corriger.

```
def dichotomie(f, a, b, eps):
    while (b - a) > eps:
        m = (a + b) / 2
        if f(a) * f(m) < 0:
            b = m
        else:
            a = m
    return (a + b) / 2
```

Exercice 3

1. L'objectif est de tester une fonction qui prend un entier naturel en paramètre et qui doit renvoyer `True` si le paramètre est un nombre premier et `False` sinon.
Pour cela, écrire une fonction `test1` qui utilise une liste de nombres premiers, comme `[2, 3, 5, 7, 11, 13, 17, 19]`, et contrôle pour tous ces nombres si la fonction passée en paramètre renvoie la bonne réponse, c'est-à-dire `True`.
2. Tester, avec la fonction `test1`, la fonction `premier` définie ci-dessous.

```
def premier(n):
    if n == 0 or n == 1:
        return False
    if n == 2:
        return True
```

```

for d in range(2, n):
    if n % d == 0:
        return False
    return True

```

3. Écrire une fonction `test2` qui exécute un contrôle pour tous les nombres non premiers compris entre 0 et 10 inclus. Utiliser cette fonction pour tester la fonction `premier`.
Ce test doit montrer que le code ne produit pas le résultat attendu.
4. Effectuer la modification dans le code de la fonction `premier` pour corriger l'erreur qui a été constatée à la question précédente.

Exercice 4

Voici un algorithme pour déterminer le plus grand élément d'un tableau de nombres non vide représenté par une liste. Prouver sa correction.

```

def maximum(tab):
    maxi = tab[0]
    for i in range(1, len(tab)):
        if tab[i] > maxi:
            maxi = tab[i]
    return maxi

```

Exercice 5

La fonction `div_euclid` doit renvoyer le quotient et le reste de la division euclidienne de m par n où m et n sont deux entiers naturels avec n non nul.

Le code de cette fonction comporte une erreur.

```

def div_euclid(m, n):
    """m et n sont deux entiers naturels, n est non nul,
    renvoie le quotient et le reste de la division de m par n
    une propriété invariante est m == n * q + r"""
    assert n != 0
    q, r = 0, m
    assert m == n * q + r
    while r > n:
        r = r - n
        q = q + 1
        assert m == n * q + r
    return q, r

```

1. Prouver la terminaison de l'algorithme.
2. Prouver que la propriété $m = nq + r$ est un invariant de la boucle. Peut-on en déduire que l'algorithme est correct ? Si ce n'est pas le cas, corriger le code.
3. Écrire un jeu de tests pour cette fonction.

Penser à vérifier les conditions sur le reste r .