



TP01/Solution

Solution 1 :

1. 1
2. 0
3. 1 (p%n vaut 4, tandis que (4 + 9) > 5 vaut 1).
4. 1 (p/n est d'abord évalué en int, ce qui fournit 1 ; puis le résultat est converti en float, avant d'être affecté à x).
5. 1.8 (p est converti en float, avant d'être divisé par le résultat de la conversion de n en float).
6. 1.9 (p est converti en float, avant d'être ajouté à 0.5 ; le résultat est divisé par le résultat de la conversion de n en float).
7. 1 (p est converti en float, avant d'être ajouté à 0.5 ; le résultat (5.5) est alors converti en int avant d'être divisé par n).
8. 25
9. 45

Solution 2 :

- A : i = 1 n = 0
 B : i = 11 n = 11
 C : i = 21 j = 6 n = 120
 D : i = 18 n = 18
 E : i = 12 j = 4 n = 12

Solution 3 :

- A : n = 10 p = 10 q = 10 r = 1
 B : n = 15 p = 10 q = 5
 C : n = 15 p = 11 q = 10
 D : n = 16 p = 11 q = 15

Solution 4 :

Il ne faut pas oublier que les opérateurs && et || n'évaluent leur second opérande que lorsque cela est nécessaire. Ainsi, ici, il n'est pas évalué dans les cas 1 et 3. Voici les résultats fournis par ce programme :

- A : n = 6 p = 2 q = 1
 B : n = 6 p = 3 q = 1
 C : n = 6 p = 2 q = 0
 D : n = 6 p = 3 q = 1

Solution 5 :

- A : i = 5, n = 5
 B : i = 5, n = 5
 C : i = 9, n = 5
 D : i = 1, n = 1
 D : i = 2, n = 3



TP01/Solution

D : $i = 3, n = 6$

E : $i = 3, n = 6$

Solution 6 :

```
#include <iostream>
using namespace std ;
main() {
    int u1, u2, u3 ; /* pour "parcourir" la suite */
    int n ;          /* rang du terme demandé */
    int i ;          /* compteur */
    do {
        cout << "rang du terme demandé (au moins 3) ? " ;
        cin >> n ;
    }
    while (n<3) ;
    u2 = u1 = 1 ;    /* les deux premiers termes */
    i = 2 ;
    while (i++ <= n) { /* attention, l'algorithm ne fonctionne */
        u3 = u1 + u2 ; /* que pour n > 2 */
        u1 = u2 ;
        u2 = u3 ;
    }
    // autre formulation possible :
    // for (i=3 ; i<=n ; i++, u1=u2, u2=u3) u3 = u1 + u2 ;
    cout << "Valeur du terme de rang " << n << " : " << u3 ;
}
```