1) Resolva a seguinte relação de recorrência.

T(1) = 1;

T(n) = T(n - 1) + 2n

T(2) = T(1) + 2* 2 = 1 + 4 = 5

T(3) = T(2) + 2* 3 = 5 + 6 = 11

T(4) =T(3) + 2* 4 = 11 + 8 = 19

$$T\,(n) = c^{n-1}\,(1) + \sum_{i=2}^{n} c^{n-'}\,g * (i)$$

$$T\,(n) = 1 + \sum_{i=2}^{n} 2i = 2 + 5 + 11 + 19 + \ldots + n$$

2) Considere a função recursiva abaixo.

```
void Proc( int n ) {
        if ( n == 0)
                return 1;
        else
                return Proc(n-1) + Proc(n-1);
}
```

1) Escreva a função recursiva que representa o número de operações da função.

$$Proc\,(0) = 1$$

$$Proc\,(n) = proc\,(n-1) + proc\,(n-1)$$

2) Resolva a recursividade encontrada.

$$Proc\,(1) = proc\,(1-1) + proc\,(1-1) = 1 + 1 = 2$$

$$Proc\,(2) = proc\,(1) + proc\,(1) = 2 + 2 = 4$$

$$Proc\,(3) = Proc\,(2) + Proc\,(2) = 4 + 4 = 8$$

$$Proc\,(4) = Proc\,(3) + Proc\,(3) = 8 + 8 = 16$$

$$Proc\,(k) = 2^k$$

Caso base:

$$Proc\,(0) = 2^0 = 1$$

Supor que:

$$Proc\,(k) = 2^k$$

Mostrar que:

$$Proc\,(k+1) = 2^{k+1}$$

$$Proc\,(k+1) = 2^k + 2^k$$

$$Proc\,(k+1) = 2^{k+1}$$

c) Diga, em relação à função computacional, o que significa resolver a função recursiva.