

## RESOLUÇÃO DO TESTE Nº2

Data: 29 out. 2018

- (1) Ambas as instruções são cíclicas. A diferença reside na sua utilização. A instrução **while** utiliza-se quando não se sabe à partida quantas iterações ou ciclos irão ser executados. No caso da instrução **for**, utiliza-se quando se sabe à partida quantas iterações ou ciclos irão ser executados.
- (2) A palavra reservada **return** serve para devolver um valor calculado numa função para uma outra que a invocou. Por exemplo, a função *int calculaNumeroAlgarismos(int)* que calcula o nº de algarismos de um nº inteiro fará uso da palavra **return** para devolver o nº de algarismos à função que chamou a função *calculaNumeroAlgarismos*.

(3) #include <stdio.h>

```
// calcula nº de algarismos do inteiro n
```

```
int size(int n)
{
    int counter=0;
    while (n>0)
    {
        n=n/10;
        counter++;
    }
    return counter;
}
```

```
int main()
{
    int N,nalgarismos,INV=0;
    printf("Introduza um nº inteiro: ");
    // lê inteiro N
    scanf("%d",&N);
    // calcula o seu nº de algarismos
    nalgarismos=size(N);
    // inverte o número N
    for(int i=1;i<=nalgarismos;i++)
    {
        INV=10*INV + N%10;
        N=N/10;
    }
    // escreve o nº invertido no ecrã
    printf("O nº invertido de % é %d\n",N,INV);
    return 0;
}
```

(4) #include <stdio.h>

```
// escreve sequência de n chars em que os chars são todos o mesmo
```

```
void charString(char ch, int n)
{
    for (int i=1; i<=n;i++)
        printf("%c",ch);
}
```

```
int main()
{
    printf("Escreva o valor de N: ");
    scanf("%d",&N);
    // inicializa nº de brancos e de asteriscos
    nbrancos = N-1;
    nasteriscos = -1;
    // para cada linha da pirâmide
    for (int i=1;i<=N;i++)
    {
        // escreve espaços em branco
        charString(" ",nbrancos);
        // escreve nº de linha
        printf("%d",i);
        // escreve asteriscos
        charString("* ",nasteriscos);
        // escreve nº de linha
        if (i!=1)
            printf("%d",i);
        // muda de linha
        printf("\n");
        // atualiza nº brancos e de asteriscos
        nbrancos--;
        nasteriscos+=2;
    }
    return 0;
}
```