

Programação Imperativa

1º Teste

1º Ano – LEI/LCC

14 de Maio de 2014 (Duração: 90 min)

1. Relativamente à função apresentada à direita,

(a) Apresente uma traçagem (simulação de todos os passos relevantes) da sua execução para um input $x = 3$ e $y = 8$ e explique qual a relação entre o input (x, y) e o output produzido pela função.

(b) Apresente uma versão não recursiva da mesma função.

```
int sp (int x, int y) {  
    if (x%2==0 && x<=y)  
        return x + sp(x+2,y);  
    else if (x%2!=0 && x<=y)  
        return sp(x+1,y);  
    else return 0;  
}
```

2. Defina uma função `int crescente (int a[], int i, int j)` que testa se os elementos do array `a`, entre as posições `i` e `j` (inclusivé) estão ordenados por ordem crescente. A função deve retornar 1 ou 0 consoante o array esteja ou não ordenado.

3. Defina uma função `int menosFreq (int v[], int N)` que recebe um array `v` com `N` elementos **ordenado por ordem crescente** e retorna o **menos frequente** dos elementos do array (se houver mais do que um elemento nessas condições deve retornar o que começa por aparecer no índice mais baixo).

4. Defina uma função `void palavra (int N, char sopa[N][N], char f[])` que, dada uma sopa de letras (representada por uma matriz de `N x N` caracteres) e um caminho (representado na string `f`) constituído pelos caracteres 'D' (direita), 'E' (esquerda), 'C' (cima) e 'B' (baixo) e terminada pelo carácter '\0', coloca na string `f` a palavra armazenada na sopa de letras, começando na posição `[0][0]` e seguindo o caminho dado.

Assuma que o caminho é válido, i.e., que não sai da matriz.

Por exemplo, se a matriz `s` for `{{'B','A','T'},{'O','T','A'},{'L','O','T'}}`, e a string `c` contiver "DDBEC", a invocação `palavra (3,s,c)` deve colocar na string `c` "BATATA".

5. Dado um vector de `N` inteiros, a mediana desse vector é um elemento `x` desse vector tal que

- no máximo $N/2$ elementos do vector são maiores do que `x` e
- no máximo $N/2$ elementos do vector são menores do que `x`.

Por exemplo, se `v` for o vector com os valores `[11 15 13 12 14 13 12 14]` a mediana será 13: há 3 elementos menores `[11 12 12]` e 3 maiores `[15 14 14]`

Defina uma função `int mediana (int v[], int N)` que calcula a mediana de um array `v` com `N` elementos.

Note que se o array estiver ordenado, a mediana corresponde ao elemento que está numa posição a meio do array.