

		%
1	7	
2	7	
3	6	

Eng. Informática

unidade curricular: **Algoritmia** ano lectivo: **2016 / 2017**

Teste Prático 02. **17.Mai.2017** duração: **60 min** (+5 min tolerância)
(sem consulta)

Número: _____ Nome: _____

Considere o seguinte programa em linguagem C:

```
typedef struct _DIR{
    char    nome[20];
    int     n_fich;
    int     n_Kb;
}DIR;

typedef struct _BTREE_NODE
{
    void * data;
    struct _BTREE_NODE * left;
    struct _BTREE_NODE * right;
} BTREE_NODE;

#define DATA(node)    ((node)->data)
#define LEFT(node)     ((node)->left)
#define RIGHT(node)    ((node)->right)
typedef BTREE_NODE * BTREE;
typedef enum _BOOLEAN {FALSE = 0,TRUE = 1} BOOLEAN;
typedef enum _STATUS {ERROR = 0,OK = 1} STATUS;

int main(int argc, char *argv[])
{
    BTREE btree;
    void *  dirs[15];
    char  file_name[20];

    printf("Nome do ficheiro: ");
    scanf("%s", file_name);
    if(ReadFile(dirs,file_name))
    {
        btree = CreateBtree(dirs,0,15);

        BtreeFree(btree);
    }
    else
        printf("ERRO na leitura do ficheiro\n");
    return 1;
}
```

```
BTREE_NODE * NewBtreeNode(void * data)
{
    BTREE_NODE * tmp_pt;
    if ((tmp_pt = (BTREE_NODE *)malloc(sizeof(BTREE_NODE)))!=NULL) {
        DATA(tmp_pt) = data;
        LEFT(tmp_pt) = RIGHT(tmp_pt) = NULL;
    }
    return tmp_pt;
}

void BtreeFree(BTREE btree)
{
    if (btree != NULL) {
        BtreeFree(LEFT(btree));
        BtreeFree(RIGHT(btree));
        free(DATA(btree));
        free(btree);
    }
}

BTREE_NODE *InitNode(void * ptr_data,BTREE_NODE * node1,BTREE_NODE * node2)
{
    BTREE_NODE * tmp_pt = NULL;

    tmp_pt = NewBtreeNode(ptr_data);
    LEFT(tmp_pt) = node1;
    RIGHT(tmp_pt) = node2;
    return(tmp_pt);
}

BTREE_NODE *CreateBtree(void ** v, int i, int size)
{
    if (i >= size)
        return(NULL);
    else
        return(InitNode(*(v+i),CreateBtree(v,2*i+1,size),CreateBtree(v,2*i+2,size)));
}
```

Sabendo que a árvore binária acima referida modela a árvore de diretorias de um PC, responda às seguintes questões desenvolvendo o código (use obrigatoriamente funções invocadas no main()) que ache necessário. Os dados de cada diretoria são: nome da diretoria, número de ficheiros e número de kbytes nela contidos.

- 01** Desenvolva o código necessário para mostrar no ecrã todos os dados das diretorias que contenham mais de 1000 Kbytes armazenados.
- 02** Desenvolva o código necessário para mostrar no ecrã o número total de ficheiros armazenado por uma diretoria (e suas subdiretorias) cujo nome deve ser lido através do teclado.
- 03** Desenvolva o código necessário para contar e mostrar no ecrã o número de diretorias vazias.