



Universidade da Beira Interior

Departamento de Informática

2009/10

1. **(2 valores)** Implemente uma macro em linguagem C que receba 2 valores inteiros e verifique se algum deles é divisor do outro. A macro deverá assumir "1" em caso verdadeiro, ou "0" caso contrário.
2. A seguinte estrutura de dados guarda as várias notas de um aluno obtidas nos trabalhos práticos numa disciplina:

```
typedef struct {
    int numero; //número do aluno
    float *notas; //vector de notas
    int totNotas; //total de notas do aluno
}Nota;
```

- a. **(2 valores)** Implemente uma função que copie integralmente o conteúdo de um elemento de "Nota" para outro, passados por referencia para a função.

```
void copia(Nota* destino, Nota *origem);
```

- b. **(2 valores)** Utilize a função implementada na alínea anterior para criar uma função que receba um vector de notas e devolva um "vector-cópia".

```
Nota* copiaVector(Nota *v, int totV);
//v=vector, totV=total de elementos
```

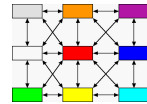
3. Considere um ficheiro de texto que guarda os tempos de corrida dos participantes numa maratona de BTT. Este tem o seguinte formato:

(#participante \t #horas \t #minutos \t #segundos \t #milisegundos)

67	4	16	23	184
167	3	17	56	266
13	2	18	13	272
...				

- a. **(3 valores)** Crie uma estrutura de dados "BTT" para guardar esta informação e implemente uma função que leia o conteúdo do ficheiro e o coloque nessa estrutura de dados.

```
BTT* leFicheiro(char *ficheiro, int *totV);
//ficheiro=nome do ficheiro a ler, totV=parâmetro de saída para o número de elementos lidos
```



- b. (2 valores) Implemente uma função que receba um vector de elementos do tipo "BTT" e mostre no écran o número do participante vencedor, bem como o respectivo tempo de prova

```
void mostraVencedor(BTT *v, int totV);  
//v=vector, totV=total de elementos no vector
```

- c. (2 valores) Crie um programa que utilize de forma conveniente as 2 funções anteriormente implementadas. Antes de terminar liberte todos os recursos dinamicamente alocados.
4. (4 valores) Implemente uma função recursiva que some os elementos de um vector de números reais que sejam maiores que o seu antecessor (elemento da posição à esquerda).

```
float somaVector(float *v, int totV);  
//v= vector; totV=total de elementos
```

5. (3 valores) Considere o seguinte programa em linguagem C. Complete as instruções de *output*, exemplifique um *input* correspondente ao seu pior caso e calcule a respectiva complexidade computacional utilizando a notação *Big-O*.

```
#include <stdio.h>  
  
int main(){  
    int *v=NULL, tot=0, i,j,a;  
    do{  
        scanf("%d",&a);  
        if (a<=0)  
            break;  
        v=(int*)realloc(v,++tot*sizeof(int));  
        v[tot-1]=a;  
    }while(1);  
    for (i=0; i<tot; i++){  
        for (j=0;j<tot;j++){  
            if (i==j)  
                continue;  
            if (v[i]%v[j]==0)  
                break;  
        }  
        if (j==tot)  
            printf("????????????????\n");  
        else  
            printf("????????????????\n");  
    }  
    free(v);  
    return(0);  
}
```