

Exame IP Época Normal

29 de Janeiro de 2016

- **2. b) e 3** – *Análise do problema, pseudocódigo e codificação em C.*
- **2. a)** – *Apenas codificação em C.*

- 2. a) [2,5 valores]** Elabore uma função que calcule a média aritmética de um conjunto de n números inteiros, de acordo com o seguinte protótipo:

```
float media_final (int valor, int valido);
```

Para o cálculo da média aritmética de n valores inteiros, a função, de cada uma das n vezes que é utilizada, recebe no primeiro parâmetro, um valor inteiro (valor), que é utilizado no cálculo do valor da média e, no segundo parâmetro (valido) um valor inteiro, que pode assumir o valor 1 ou 0 e que determina o comportamento da função.

Assim, quando o segundo argumento da função for igual a 1 (valido=1), a função adiciona o primeiro argumento (valor) ao cálculo da média (e devolve 0). Caso contrário, se o segundo argumento da função for igual a 0 (valido=0), a função não adiciona o primeiro argumento (valor) ao cálculo da média e devolve o valor da média aritmética dos n valores inteiros anteriormente considerados.

b) [4,0 valores] Recorrendo à função desenvolvida na alínea **a)**, elabore um programa que, dado um valor inteiro n (superior a 0 e não superior a 10), calcule a média aritmética dos valores inteiros positivos existentes no conjunto de n valores inteiros obtidos do utilizador. O resultado (a média calculada) deve ser apresentado com dois dígitos decimais.

- 3. [5,5 valores]** Escreva uma função que receba como argumentos duas *strings*, **st1** e **st2**, bem como um *array* de caracteres **stfim**, e um inteiro **tamf** correspondente à dimensão deste vetor.

A função deverá construir uma nova *string* em **stfim** com os caracteres alternados das duas *strings* **st1** e **st2**. Ou seja, os dois primeiros caracteres são, respetivamente, o primeiro carácter da **st1** e o primeiro carácter da **st2**; o terceiro e o quarto são, respetivamente, o segundo carácter da **st1** e o segundo carácter da **st2**, e assim sucessivamente.

A função deve devolver 1 se **stfim** for de facto modificada, e 0 caso contrário.

O protótipo da função será:

```
int func (char st1[], char st2[], char stfim[], int tamf);
```

Considerando a seguinte função main():

```
#include <stdio.h>
#define TAMFRASE 50
#define TAMFIM 75
void main()
{
    char f1[TAMFRASE], f2[TAMFRASE], final[TAMFIM]="";
    printf("Indique a frase 1: "); gets(f1);
    printf("Indique a frase 2: "); gets(f2);
    if(func(f1,f2,final,TAMFIM))
        printf("Frase modificada:\n%s\n", final);
    else
        printf("Frase não modificada!\n");
}
```

... o resultado da execução deverá ser o seguinte:

```
Indique a frase 1: Vamos jogar bilhar...
Indique a frase 2: LOL Quando!
Frase modificada:
VLaoMLo sQ ujaongdaor! bilhar...
```

Nota: Caso necessite pode utilizar a função:

```
int strlen(char str[]);
```

a qual devolve o número de caracteres efetivos existentes na *string* "str" (sem contar com o '\0' final).

Proposta de Resolução

2. a)

```
float media_final(int valor, int valido)
{
    static int nvalidos=0, soma=0;
    /*Usam-se variáveis estáticas porque estas só são inicializadas
    da primeira vez que a função é executada. Assim, mantêm o mesmo
    valor de uma execução da função para a execução seguinte.*/

    if(valido==1){
        nvalidos+=1;
        soma+=valor;
        return 0;
    }
    return (float)soma/nvalidos;
}
```

2. b)

```
#include <stdio.h>

float media_final(int valor, int valido);
void main()
{
    int i, n, valor, validade;
    float media;

    printf("Introduza o numero de valores que pretende
    analisar (entre 1 e 10)\n");

    do{
        scanf("%d",&n);

        if(n<=0 || n>10)
            printf("Valor invalido!\n");
    } while(n<=0 || n>10);

    for(i=0;i<n;i++){
```

```

        printf("Valor %d? ", i+1);
        scanf("%d",&valor);

        if(valor>0)
            validade=1;
        else validade=0;

        media=media_final(valor,validade);
    }

    if(media)
        printf("Media final: %.2f\n",media);
    else
        printf("Nao foram introduzidos valores validos
suficientes!\n");
}

```

3.

```

int func(char st1[], char st2[], char stfim[], int tamf)
{
    int i, j, tamf1, tamf2;

    tamf1=strlen(st1);
    tamf2=strlen(st2);

    if(tamf>=tamf1+tamf2){
        if(tamf1>tamf2){
            for(i=0,j=0;st2[j]!='\0';j++){
                stfim[i]=st1[j];
                stfim[i+1]=st2[j];

                if(j==tamf2)
                    i++;
                else i+=2;
            }
            for(;st1[j]!='\0';j++,i++)
                stfim[i]=st1[j];

```

```

        stfim[i]='\0';
    }
    else if (tamf2>tamf1) {
        for(i=0,j=0;st1[j]!='\0';j++,i+=2) {
            stfim[i]=st1[j];
            stfim[i+1]=st2[j];
        }
        for(;st2[j]!='\0';j++,i++)
            stfim[i]=st2[j];

        stfim[i]='\0';
    }

    else{
        for(i=0,j=0;st2[j]!='\0';j++,i+=2) {
            stfim[i]=st1[j];
            stfim[i+1]=st2[j];
        }
        stfim[i]='\0';
    }

    return 1;
}

return 0;
}

```