

UNIVERSIDADE DA BEIRA INTERIOR

Programação – LEI + LMA

1º Semestre

Frequência 2

Resolução (sugestão)

04/01/2023

1.

Considere as seguintes declarações de variáveis:

```
int **V, *W, *X;
```

e que $\text{sizeof}(\text{int}) = 4$ e $\text{sizeof}(\text{int} *) = 8$.

(esquema de um bloco de memória)

	...	
100500	110160	W
	...	
100700	110168	X
	...	
110160	50	
110164	100	
110168	150	
110172	600	
110176	700	
110180	800	
	...	
110500	100700	V
	...	

Usando os valores que constam no esquema de um bloco de memória que se encontra ao lado, indique os valores de cada uma das seguintes expressões:

- a) $\&X$
- b) $X[1]$
- c) $W + 4$
- d) $W[4]$
- e) $*W + 4$
- f) $*(W + 2)$
- g) $*V + 4$
- h) $V + 4$

NOTA: apresente todos os cálculos efetuados

- a) $\&X$ = **100700**
- b) $X[1]$ = **600**
- c) $W + 4$ = $110160 + 4 \times 4 =$ **110176**
- d) $W[4]$ = **700**
- e) $*W + 4$ = $50 + 4 =$ **54**
- f) $*(W + 2)$ = $*(110160 + 2 \times 4) =$ **150**
- g) $*V + 4$ = $110168 + 4 \times 4 =$ **110184**
- h) $V + 4$ = $100700 + 4 \times 8 =$ **100732**

2.

Implementar um **subprograma** em C que dado um número inteiro positivo N (parâmetro do subprograma), peça ao utilizador para inserir N números reais e **determine e devolva** como resultados a **soma** dos números **positivos** (> 0) inseridos e a quantidade de números **positivos** (> 0) inseridos.

```
// versão 1: devolver/passar os dois resultados como parâmetros do subprograma
```

```
void numeroSomaPositivos (int N, float *soma, int *cont){
```

```
int k;
```

```
float X;
```

```
*cont = 0;
```

```
*soma = 0;
```

```
for (k = 1; k <= N; k++){
```

```
    printf("Insira um numero real: ");
```

```
    scanf("%f", &X);
```

```
    if (X > 0){
```

```
        *cont = *cont + 1;
```

```
        *soma = *soma + X;
```

```
    }
```

```
 }
```

```
}
```

```
// versão 2: devolver/passar um resultado pelo return e o outro como parâmetro do subprograma
```

```
int numeroSomaPositivos (int N, float *soma){
```

```
int k, cont;
```

```
float X;
```

```
cont = 0;
```

```
*soma = 0;
```

```
for (k = 1; k <= N; k++){
```

```
    printf("Insira um numero real: ");
```

```
    scanf("%f", &X);
```

```
    if (X > 0){
```

```
        cont = cont + 1;
```

```
        *soma = *soma + X;
```

```
    }
```

```
 }
```

```
return cont;
```

```
}
```

3.

Implementar um **subprograma** em C que dados um array (de 1 dimensão) **X** com **N** números reais (**X** e **N** são parâmetros do subprograma), **determine e devolva** o **menor** número **positivo de X**. Caso não haja positivos em **X**, o subprograma deve devolver um número negativo.

```
float menorPositivo (float X[], int N){
    float menor;
    int k;
    menor = -1;
    k = 0;
    while (k < N && X[k] <= 0)
        k++;
    if (k < N)
        menor = X[k];
    for (k; k < N; k++){
        if (X[k] > 0 && X[k] < menor)
            menor = X[k];
    }
    return menor;
}
```

4.

Considere o tipo de dados seguinte (registro/estrutura):

```
typedef struct {  
    int numAluno;  
    int notaFinal;  
    int aprovado; // 1 = aprovado, 0 = reprovado  
} ALUNO;
```

Considere também o seguinte subprograma já implementado (incluído na biblioteca **Frequencia2.h**):

```
void lerArray1DALUNO (ALUNO A[], int *N)  
    // constrói um array A com "N" registros de ALUNO, mas apenas com os campos numAluno e  
    notaFinal de cada elemento com valores atribuídos (campo aprovado sem valor atribuído)  
void mostrarArray1DALUNO (ALUNO A[], int N)  
    // mostra no monitor os N elementos do tipo ALUNO do array A
```

Implemente um **programa em C** que realize as seguintes ações (pela ordem indicada):

- ler um array **A** de registos do tipo **ALUNO** com **TAM** ($TAM \leq 100$) elementos (usar subprograma dado)
- atribuir ao campo **aprovado** de cada um dos elementos do array **A** o valor **1** (se $notaFinal \geq 10$) ou **0** (se $notaFinal < 10$)
- mostrar todos os elementos do array **A**

```
#include <stdio.h>  
typedef struct {  
    int numAluno;  
    int notaFinal;  
    int aprovado; // 1 = aprovado, 0 = reprovado  
} ALUNO;  
#include "Frequencia2.h"  
main(){  
    ALUNO A[100];  
    int TAM, k;  
    do{  
        printf("Insira um inteiro entre 1 e 100: ");  
        scanf("%d", &TAM);  
    }while(TAM < 1 || TAM > 100);  
    lerArray1DALUNO(A, &TAM); // ATENÇÃO  
    for (k = 0; k < TAM; k++)  
        if (A[k].notaFinal >= 10)  
            A[k].aprovada = 1;  
        else  
            A[k].aprovado = 0;  
    mostrarArray1DALUNO(A, TAM);  
}
```

5.

Considere-se um ficheiro de texto de nome "entrada.txt", no qual cada uma das linhas contém 2 números: um inteiro e um real (por esta ordem).

Construa um **programa em C** que leia os números de cada uma das linhas do ficheiro "entrada.txt" e guarde-os no ficheiro de texto de nome "saida.txt", mas apenas aqueles em que o **número inteiro** é **positivo (> 0)**.

NOTAS:

1. Não pode usar arrays
2. Na abertura dos ficheiros não é necessário verificar se a houve algum erro na abertura

```
#include <stdio.h>
int main(){
    int num1;
    float num2;
    FILE *f, *g;
    f = fopen("entrada.txt", "r");
    g = fopen("saida.txt", "w");
    while (fscanf(f, "%d%f", &num1, &num2) == 2)
        if (num1 > 0)
            fprintf(g, "%d %f", num1, num2);
    fclose(f);
    fclose(g);
}
```