

1 - Considere o programa em baixo. Pretende-se que o programa imprima os valores de **var1** e **var2** e os seus endereços (guardados em **ptr1** e **ptr2**).

```
#include <stdio.h>
main()
{
    int var1 = 5;
    char var2 = 'a';
    int *ptr1 = &var1;
    char *ptr2;
    *ptr2 = 'b';
    printf("var1 tem o endereço %p e o valor %d\n", ptr1, *ptr1);
    printf("var2 tem o endereço %p e o valor %d\n", ptr2, *ptr2);
}
```

- Se tentar compilar e executar este programa ocorrerá o erro “Segmentation fault”. Corrija o programa para que isso não aconteça.
- Altere o programa para que este imprima também o tamanho dos tipos de variáveis `char`, `char*`, `int` e `int*`.
- Altere o programa para que este imprima ainda os valores de **`ptr1+1`** e **`ptr2+1`** e compare com os valores de **`ptr1`** e **`ptr2`**.

```
#include <stdio.h>
main()
{
    int i, j, *p_1, *p_2, **p_p_1, **p_p_2;
    i = 4;
    j = 5;
    p_1 = &i;
    p_2 = &j;
    p_p_1 = &p_1;
    p_p_2 = &p_2;
}
```

Variável	i	j	p_1	p_2	p_p_1	p_p_2
Conteúdo	4	5				
Endereço	1000	1007	1030	1053	1071	1079

[illegible]

3 - Escreva uma função chamada **ordena** que recebe três valores e os ordena por ordem crescente. Teste a sua função com um programa que pede ao utilizador três números, invoca a função ordena e depois imprime o resultado no ecrã.

Considere o seguinte protótipo para a função:

```
void ordena(int *valorA, int *valorB, int *valorC);
```

*Exemplo:*

Insira os valores a ordenar: 43 65 17

Valores a, b, c ordenados por ordem crescente: 17 43 65

4 – Escreva uma função que recebe dois valores inteiros por referência e devolve por retorno o endereço do valor maior. Escreva um programa para testar a sua função.

Considere o seguinte protótipo para a função:

```
int *vmaior(int *valor1, int *valor2);
```

*Exemplo:*

Insira dois valores: 56 32

Endereco do maior 0016F838 e conteudo 56

5 - Implemente uma função (procedimento) que calcule o quociente e o resto da divisão inteira de dois números inteiros. Utilizando o procedimento que definir, implemente uma função que determina a soma dos dígitos de um número inteiro. Deverá definir as seguintes funções:

```
void quociresto(int dividendo, int divisor, int *quociente, int *resto);  
int soma(int n);
```

Escreva um programa que lhe permita testar as funções que desenvolveu.

6 – Escreva uma função chamada **horasMin** que converte um valor em minutos para horas:minutos. A sua função deve ainda retornar 1 se o número de horas:minutos for superior a um dia e 0 caso contrário. Para testar a sua função escreva um programa que usa a função horasMin para efetuar a conversão e que depois imprime o resultado no ecrã.

*Exemplo:*

Insira o total de minutos: 568

568 minutos correspondem a 09h:28min e não é superior a 1 dia!

Insira o total de minutos: 4689

4689 minutos correspondem a 78h:09min e é superior a 1 dia!

7 – Escreva uma função que faz a conversão entre o espaço de cor RGB e o espaço de cor HSV. A conversão entre espaços de cor é efetuada da seguinte maneira:

1. Converter os valores RGB que estão na gama [0...255] para a gama [0...1]
2. Encontrar o valor máximo (max) e mínimo (min) das componentes RGB
3. Aplicar as seguintes equações

$$H = \begin{cases} 60 \times \frac{G-B}{\max - \min}, & \text{se } \max = R \text{ e } G \geq B \\ 60 \times \frac{G-B}{\max - \min} + 360, & \text{se } \max = R \text{ e } G < B \\ 60 \times \frac{B-R}{\max - \min} + 120, & \text{se } \max = G \\ 60 \times \frac{R-G}{\max - \min}, & \text{se } \max = B \end{cases}$$

$$S = \frac{\max - \min}{\max}$$

$$V = \max$$

$$\text{Nota: } R, G, B \in [0;255]; H \in [0;360]; S \in [0;1]; V \in [0;1]$$

Assuma o seguinte protótipo para a função:

```
void rgb2hsv(int R, int G, int B, float *H, float* S, float *V);
```

Escreva um programa para testar a sua função.

*Exemplo:*

Insira as componentes RGB: 123 234 80

RGB=(123, 234, 80) <=> HSV=(103.247, 0.658, 0.918)