

# Aula prática #8 – Apontadores e Funções

## Problema 1

---

Defina e implemente uma função que calcule a divisão inteira entre dois números, sem recorrer aos operadores / e %. A função deve ter as seguintes características:

- Entradas: dividendo, divisor
- Saídas: quociente, resto
- Retorno: código de erro - 1 (sem erro) ou 0 (erro porque divisor=0)

Implemente ainda um programa que permita testar a função desenvolvida.

## Problema 2

---

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**).

```
1  #include <stdio.h>
2  int main()
3  {
4      int var1 = 5;
5      char var2 = 'a';
6      int *ptr1 = &var1;
7      char *ptr2;
8      *ptr2 = 'b';
9      printf("var1 tem o endereço %p e o valor %d\n", ptr1, *ptr1);
10     printf("var2 tem o endereço %p e o valor %d\n", ptr2, *ptr2);
11 }
```

Siga os seguintes passos e interprete para cada caso os valores apresentados pelo programa:

- 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 **ptr1+1** e **ptr2+1** e compare com **ptr1** e **ptr2**.

## Problema 3

Dado o seguinte programa complete as tabelas:

```

1  #include <stdio.h>
2  int main()
3  {
4      int i, j, *p_1, *p_2, **p_p_1, **p_p_2;
5      i = 4;
6      j = 5;
7      p_1 = &i;
8      p_2 = &j;
9      p_p_1 = &p_2;
10     p_p_2 = &p_1;
11 }

```

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

Expressão	i	*p_2	&i	&p_2	**p_p_1	*p_p_2	&(*p_1)	j	*p_1	*(&p_1)
Resultado										

## Problema 4

---

Escreva e teste uma biblioteca de funções chamada `polinomios` para operar com polinómios. Considere que os coeficientes dos polinómios são inteiros e que o grau máximo é 10.

a) Implemente uma função de leitura de polinómios com a seguinte assinatura:

```
1 int lerPoli(int *poli)
```

b) Implemente uma função de escrita de polinómios com a seguinte assinatura:

```
1 void escrevePoli(int *poli, int grau)
```

c) Implemente uma função que adicione 2 polinómios com a seguinte assinatura:

```
1 void adicionaPoli(int *p1, int *p2, int *pRes, int maiorGrau)
```

d) Implemente uma função que calcule o valor do polinómios para um certo valor da variável independente com a seguinte assinatura:

```
1 float calc(int *poli, int grau, float x)
```

e) Implemente um programa que permita testar a biblioteca desenvolvida.

## Problema 5

---

Implemente a função **contaVezes** que conta quantas vezes um determinado dígito existe num número pedido ao utilizador. A função não deve retornar nenhum valor, em vez disso, o resultado deve ser guardado utilizando o apontador **ptr\_vezes**.

```
1 void contaVezes(int num, int dig, int *ptr_vezes);
```

### Exemplo

```
1 Introduza um numero inteiro: 3276022
2 introduza um digito: 2
3 O digito 2 aparece 3 vezes no numero 3276022
```

## Problema 6

**Nota:** Este problema surge na aula prática 5, mas é aqui repetido como introdução ao problema 7.

O seno de  $x$  pode ser calculado usando o desenvolvimento em série de Taylor:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots + (-1)^{n+1} \frac{x^{2n-1}}{(2n-1)!}, n = 1, 2, 3, \dots$$

Escreva e teste uma função que determine o seno de um ângulo  $x$  (introduzido pelo utilizador em radianos), usando este desenvolvimento em série de Taylor, até que o valor absoluto de um termo esteja abaixo de uma tolerância especificada pelo utilizador. Esse último termo já não é incluído no cálculo do seno.

```
1 float seno(float x, float tolerancia);
```

Sugestão: utilize outra uma função para calcular o fatorial de um número, semelhante à apresentada nas aulas teóricas.

### Exemplo

```
1 Qual o valor de x? 1.57
2 Qual o valor da tolerancia? 0.00005
3 O seno de 1.57 e' 1.000003
```

**Nota:** Repare que  $1.57 \approx \pi/2$ .

## Problema 7

Altere o último programa para imprimir também o número de termos utilizados no cálculo do seno de  $x$ . Para isso, altere a função seno para devolver este número por referência:

```
1 float seno(float x, float tolerancia, int *termos);
```

### Exemplo

```
1 Qual o valor de x? 1.57
2 Qual o valor da tolerancia? 0.00005
3 O seno de 1.57 e' 1.000003 (5 termos da serie)
```