

Aula prática #7 – Funções e Apontadores

Problema 1

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:

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

Exemplo

```
1  Insira dois valores: 56 32
2  Endereco do maior: 0x0016F838
3  Valor: 56
```

Problema 2

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:

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

Exemplo

```
1  Insira os valores a ordenar: 43 65 17
2  Valores a, b, c ordenados por ordem crescente: 17 43 65
```

Problema 3

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.

```
1  int horasMin(int totalMins, int *hours, int *minutes);
```

Exemplo

```
1  Insira o total de minutos:568
2  568 minutos correspondem a 09h:28m e nao e' superior a 1 dia.
3
4  Insira o total de minutos:4689
5  4689 minutos correspondem a 78h:09m e e' superior a 1 dia.
```

Problema 4

Implemente duas funções que convertam coordenadas cartesianas em coordenadas polares e vice-versa. Deverá definir as seguintes funções:

```
1  void cartesianas_polares(float x, float y, float *r, float *theta);
2  void polares_cartesianas(float r, float theta, float *x, float *y);
```

As relações entre coordenadas cartesianas em coordenadas polares são definidas por:

$$x = r \cdot \cos \theta$$

$$y = r \cdot \sin \theta$$

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \text{atan2}(y, x)$$

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

Problema 5

Implemente um procedimento que calcule o quociente e o resto da divisão inteira de dois números inteiros, sem utilizar os operadores % e /. Utilizando o procedimento que definir, implemente uma função que determina a soma dos dígitos de um número inteiro e outra função que teste se um número é par. Deverá definir as seguintes funções:

```
1 void quociresto(int dividendo, int divisor, int *quociente, int *resto);
2 int soma(int n);
3 int par(int n);
```

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

Problema 6

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 endereco %p e o valor %d\n", ptr1, *ptr1);
10    printf("var2 tem o endereco %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 7

Dado o seguinte programa complete as tabelas:

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

```

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 }

```

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

Problema 8

Escreva uma função que conte quantas vezes um determinado dígito existe num número pedido ao utilizador.

Exemplo

```

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

```

Problema 9

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

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

Escreva e teste uma função que determine o seno de um ângulo x (introduzido pelo utilizador em radianos), usando este desenvolvimento em série, 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 na aula teórica 9.

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 10

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)
```

Problema 11

Uma resistência é um componente passivo de 2 terminais utilizado na maioria dos circuitos eletrônicos. Implemente um conversor que transforme um código de cores no valor da resistência em Ω . Para tal complete o código disponível em **prob11.c**, em particular o procedimento seguinte:

```
1 int converte_codigo_cores(int cor, int pos, float *ret);
```

Este procedimento tem como argumentos um inteiro representando a cor de cada banda da resistência, um inteiro representando a posição dessa cor (1ª, 2ª ou 3ª banda) e um apontador para float contendo o valor atual da resistência. Não considere a tolerância. Cada cor é representada por um número inteiro de 0 a 9, de acordo com <https://www.digikey.sg/en/resources/conversion-calculators/conversion-calculator-resistor-color-code-4-band>, reproduzido em seguida:

0 - Black; 1 - Brown; 2 - Red; 3 - Orange; 4 - Yellow; 5 - Green; 6 - Blue; 7 - Violet; 8 - Grey; 9 - White.

A função deverá ainda retornar 1 em caso de sucesso e -1 em caso de erro (valor da posição ou da resistência inválidos).

Exemplo

```
1 Introduza a cor da banda 1: 0
2 Introduza a cor da banda 2: 1
3 Introduza a cor da banda 3: 2
4 Valor da resistencia: 100.00 Ohms
```