

Aula prática #7 – Funções

Problema 1

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, até um termo em que a diferença para o termo seguinte tenha um valor absoluto abaixo de uma tolerância (especificada pelo utilizador). Sugestão: utilize 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: Note que $1.57 = \pi/2$.

Problema 2

Escreva uma função determina se um número b está contido num número a , ambos pedidos ao utilizador:

Exemplo

```
1 Introduza um numero: 567890
2 Introduza um segundo numero: 678
3 O numero 678 esta contido no numero 567890
4 Introduza um numero: 123456
5 Introduza um segundo numero: 321
6 O numero 321 nao esta contido no numero 123456
```

Problema 3

Implemente um programa que use o conceito de funções para calcular o valor das seguintes séries, com n a especificar pelo utilizador:

$$Q(n) = 1 + \frac{1}{4} + \frac{1}{9} + \dots + \frac{1}{n^2}$$
$$A(n) = \frac{1}{2} - \frac{1}{2^2} + \frac{1}{2^3} - \frac{1}{2^4} + \dots + (-1)^{n+1} \frac{1}{2^n}$$

Exemplo

```
1 Indique o valor de n? 4
2 Q(4)=1.42 e A(4)=0.3125
```

Problema 4

Desenvolva um programa que possa ser utilizado por alunos do ensino básico para aprender a somar e a multiplicar (inteiros). Utilize a função `rand()` para gerar dois números inteiros positivos (menores do que 100) e selecionar a operação. O seu programa deve depois apresentar a pergunta

Exemplo

```
1 Quanto e' 6 vezes 7?
```

O aluno introduz a resposta e o seu programa verifica a resposta. Caso a resposta esteja correta deve escrever a mensagem “Muito Bem!” e deve apresentar uma nova questão. Caso esteja errada deve escrever a mensagem “Está errado, tenta outra vez!” e apresentar novamente a mesma questão. O seu programa deve guardar um registo do número de questões respondidas corretamente à primeira.

Problema 5

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

Exemplo

```
1 Introduza um numero inteiro: 37628392022
2 introduza um digito: 2
3 O digito 2 aparece 4 vezes no numero 37628392022
```

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  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 os valores de **ptr1+1** e **ptr2+1** e compare com os valores de **ptr1** e **ptr2**.

Problema 7

Dado o seguinte programa complete as tabelas:

```

1  #include <stdio.h>
2  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
Conteúdo	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)
Conteúdo										

Problema 8

Com este exercício pretende-se descobrir e corrigir bugs num programa usando o **GDB**. Para tal iremos usar um programa que usa uma função para converter letras minúsculas em maiúsculas e vice-versa. No entanto, devido à existência de bugs, o programa não desempenha corretamente essa função.

Nota: Escreva uma para converter letras minúsculas em maiúsculas e vice-versa.

8.1 – Compile e corra o seguinte programa. Este programa imprime “:)” caso o programa se encontre correto e “:(” caso contrário. Conforme esperado, o programa deverá imprimir “:(”.

```
1  #include<stdio.h>
2
3  void converte(char c)
4  {
5      int diff = 'a' - 'z';
6      if (c >= 'A' || c <= 'Z')
7          c = c + diff;
8      else
9          c = c - diff;
10
11     if(c=='A')
12         printf(":\n");
13     else
14         printf(":(\n");
15 }
16
17 int main()
18 {
19     char c='a';
20     converte(c);
21     return 0;
22 }
```

8.2 – Use o **GDB** para descobrir a localização dos bugs.

8.3 – Corrija os bugs. Depois de os bugs ficar devidamente corrigido, o programa deverá imprimir “:)”.

8.4 – Altere o código de forma a permitir a introdução de qualquer letra e imprima a conversão.