Aula prática #6 – Funções Básicas

Problema 1

Considere o problema 3 da ficha 5 (lançamento de um dado).

1.1 – Reescreva o programa usando o conceito de funções. Deverá usar a seguinte função, que retorna um número aleatório entre 1 e 6:

```
1 int dado();
```

1.2 – Reescreva o programa, implementando a seguinte função, que retorna um número inteiro aleatório entre os limites inferior e superior recebidos como parâmetros:

```
1 int aleatorio(int limiteInferior, int limiteSuperior);
```

Problema 2

Escreva um programa que calcule o peso ideal de uma pessoa (em quilos) sabendo que para homens, $pesoideal = 72.7 \times altura - 58$, e para mulheres, $pesoideal = 62.1 \times altura – 44.7$. O cálculo deverá ser feito por uma função, que recebe como argumentos a altura (em metros) e o sexo da pessoa.

Exemplo

```
1 Insira a altura da pessoa: 1.75
2 Insira o sexo da pessoa(M/F): M
3 O peso ideal seria de 69.22 quilos
```

Problema 3

Escreva um programa que leia dois números da consola e imprima o resultado das operações soma, diferença, multiplicação, divisão e módulo entre os dois números. Para isso, deve usar um procedimento que recebe dois inteiros e imprima na consola os resultados.

Exemplo

```
Introduza dois numeros: 5 10
5 + 10 = 15
5 - 10 = -5
5 * 10 = 50
5 / 10 = 0.5
5 \% 10 = 5
10 + 5 = 15
10 - 5 = 5
10 * 5 = 50
10 / 5 = 2.0
10 \% 5 = 0
```

Problema 4

Implemente uma função, com parâmetros a e x, para o cálculo de $f(x) = a \times x^2$ (parábola). Utilize a função num programa que apresenta os valores de $f(x)$ para valores de x num determinado intervalo definido pelo utilizador. O utilizador deve especificar os limites (inferior e superior) do intervalo, bem como o incremento a utilizar.

Exemplo

```
Qual o valor de a? 2
Qual o intervalo? 1 2
Qual o incremento? 0.5
f(1.0)=2.0
f(1.5)=4.5
f(2.0)=8.0
```

Problema 5

Escreva um programa que desenhe um retângulo, através de um procedimento ao qual são passados três parâmetros: caráter a utilizar, número de linhas e número de colunas.

Exemplo

```
Introduza um carater: x
Introduza o numero de linhas: 4
Introduza o numero de colunas: 6
xxxxxx
x____x
x____x
xxxxxx
```

Problema 6

Implemente um procedimento que determine as soluções reais do tipo $ax^2 + bx + c = 0$. O procedimento deve estar preparado para detectar raízes complexas.

Nota: A

inclusão da biblioteca matemática externa obriga a acrescentar "-lm" no final da instrução de compilação do programa.

Exemplo

```
Introduza o valor de a: 2
Introduza o valor de b: 4
Introduza o valor de a: 5
Tem raizes complexas: -1+1.2247i e -1-1.2247i

Introduza o valor de a: 1
Introduza o valor de b: 2
Introduza o valor de a: 1
Tem uma raiz dupla: -1

Introduza o valor de a: 1
Introduza o valor de b: 4
Introduza o valor de a: 1
As raizes sao: -0.27 e -3.73
```

Problema 7

Com este exercício pretende-se descobrir e corrigir bugs num programa usando o **GDB**. Para tal iremos usar um programa que determina se um número é perfeito. No entanto, devido à existência de bugs, o programa não desempenha corretamente essa função.

Nota: Escreva um programa que determina se um número é perfeito.

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

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

```c
#include<stdio.h>

void perfeito(int n)
{
  int soma=0, i;
  for(i=0;i<=n;i++)
  {
    if(n/i!=0)
      soma+=i;
  }
  if(soma==n)
    printf(":)\n");
  else
    printf(":(\n");
}

int main()
{
  int n;
  n = 28;
  perfeito(n);
   return 0;
}
```

7.3 – Corrija os bugs. Depois de os bugs ficar devidamente corrigido, o programa deverá imprimir ":)".

7.4 – Altere o código de forma a permitir o teste de qualquer número inteiro.

Nota: Os seguintes números são perfeitos: 6, 28, 496, 8128.