

# Aula prática #4 – Estruturas de Controlo (Repetição)

Para mais desafios consulte <http://projecteuler.net/problems>.

## Problema 1

---

Escreva um programa que leia um número inteiro e calcule a soma dos seus dígitos.

```
1  Insira um numero? 325
2  A soma dos digitos e 10.
```

## Problema 2

---

**2.1** — Escreva um programa que calcule a média, o máximo e o mínimo de uma lista de números lidos. O programa deve ler valores até que função “scanf” retorne um código de erro. Para indicar o fim da introdução dos dados basta introduzir “<Ctrl+D>” (tecla “control” + tecla “D”) no início de uma nova linha.

```
1  4
2  2
3  7
4  8
5  <Ctrl+D>
6  A media dos numeros e 5.25, o maximo e 8 e o minimo e 2.
```

O exemplo que se segue mostra uma possível forma de atingir este requisito.

```
1  #include <stdio.h>
2
3  int main() {
4      float num;
5      while(1) {
6          if(scanf("%f", &num) != 1) { //scanf retorna o numero de elementos lidos com sucesso
7              break;
8          }
9          .....
10     }
11 }
```

**2.2** – Crie um ficheiro com uma lista de números e execute o seu programa com os dados constantes nesse mesmo ficheiro utilizando o redirecionamento da entrada (usando < no terminal).

```
1 $ cat < ex2.txt
2 4
3 2
4 7
5 8
6 <Ctrl+D>
7 $ ./a.out < ex2.txt
8 A media dos numeros e 5.25, o maximo e 8 e o minimo e 2.
```

## Problema 3

Escreva um programa que determinar o capital acumulado ( $c_a$ ) ao fim de  $n$  anos de um capital inicial ( $c_i$ ) atualizado a uma taxa de juro anual constante ( $j$ ). Os valores de  $c_i$ ,  $j$  e o número de anos são especificados pelo utilizador.

**Nota:** A taxa de juro é calculada com a seguinte formula:  $c_a = c_i * \left(1 + \frac{j}{100}\right)^n$

```
1 Qual o capital inicial? 200
2 Qual a taxa de juro dada pelo banco? 5
3 Durante quantos anos? 3
4 O capital acumulado ao fim de 3 anos e de 231.525 euros
```

## Problema 4

Construa um programa que imprima uma árvore de Natal em formato “ASCII”. A árvore é caracterizada pela sua altura (número de linhas, igual ou superior a 4), e deve estar corretamente alinhada.

```
1 altura? 6
2  ____*
3  ____***
4  ____*****
5  _*****
6  *****
7  ____*
```

## Problema 5

---

Escreva um programa que leia um número inteiro e o fatorize. Recorra ao método mais simples de fatorização, verificando sempre todos os inteiros maiores que 1 e menores que o número a fatorizar.

```
1  Insira um numero? 25830
2  2 3 3 5 7 41
```

## Problema 6

---

Escreva um programa que calcule o checksum digit do sistema EAN-13. Mais informações em: [http://en.wikipedia.org/wiki/EAN-13#Calculation\\_of\\_checksum\\_digit](http://en.wikipedia.org/wiki/EAN-13#Calculation_of_checksum_digit)

```
1  Insira um EAN-13 (sem checksum): 400638133393
2  1
3  Insira um EAN-13 (sem checksum): 503848350048
4  4
```

## Problema 7

---

Escreva um programa que faça uma multiplicação entre um qualquer número e outro que seja potência de base 2 sem usar o operador de multiplicação "\*".

**Sugestão:** Use o operador bitwise left shift "<<" que afeta o número na base binária. Alguns exemplos da utilização deste operador:

- $3 \ll 1 = 6;$
- $3 \ll 2 = 12;$
- $2 \ll 4 = 32.$

```
1  Insira um operando: 3
2  Insira outro operando (potencia de base 2): 8
3  O resultado da multiplicacao e: 24
4  Insira um operando: 5
5  Insira outro operando (potencia de base 2): 4
6  O resultado da multiplicacao e: 20
```

## Problema 8

Com este exercício pretende-se descobrir e corrigir um bug num programa usando o **GDB**. Para tal iremos usar um programa cujo objetivo seria calcular o valor de  $\pi$  usando a seguinte relação:

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \frac{4}{13} - \dots \quad (1)$$

No entanto, devido à existência de um bug, o valor de  $\pi$  não é calculado corretamente.

**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  #include <math.h>
3
4  #define PRECISION 0.00001
5
6  float pi(int iter) {
7      double v = 0;
8
9      for(int i = 0; i < iter; i++) {
10         double tmp = 4 / (4 * i + 1) - 4 / (4 * i + 3);
11         v += tmp;
12     }
13
14     return v;
15 }
16
17 int main() {
18     float v = pi(1000000);
19
20     puts((fabs(v - M_PI) < PRECISION) ? ":)" : ":(");
21
22     return 0;
23 }
```

**8.2** – Use o **GDB** para descobrir a localização do bug.

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