

# Aula prática #3 – Estruturas de Controlo (Seleção)

## Problema 1

---

Implemente um programa que identifique se um número é par ou ímpar.

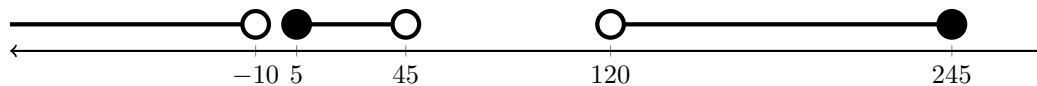
### Exemplo

```
1 Insira um numero: 50
2 50 e par.
3 Insira um numero: 25
4 25 e impar.
```

## Problema 2

---

Implemente um programa que indique se um número solicitado ao utilizador se encontra em algum dos seguintes intervalos:



### Exemplo

```
1 Indique um numero: 50
2 O numero nao se encontra nos intervalos
3 Indique um numero: 200
4 O numero encontra-se nos intervalos
```

## Problema 3

---

Escreva um programa que converta informação quantitativa em informação qualitativa sobre a altura das pessoas. Utilize as seguintes regras:

- uma pessoa com menos de 1.3 m é baixíssima;
- uma pessoa com altura entre 1.3 m e 1.6 m é baixa;
- uma pessoa com altura entre 1.6 m e 1.75 m é mediana;
- uma pessoa com altura entre 1.75 m e 1.9 m é alta;
- uma pessoa com altura superior a 1.9 m é altíssima.

### Exemplo

```
1 Qual a altura da pessoa? 1.7
2 Essa pessoa e Mediana.
```

## Problema 4

---

Escreva um programa que verifique se um número de três dígitos é capicua.

### Exemplo

```
1 Insira um numero de 3 digitos: 232
2 O numero e capicua
3 Insira um numero de 3 digitos: 332
4 O numero nao e capicua
```

## Problema 5

---

Crie um programa que simule uma máquina de calcular que efetua as operações aritméticas mais básicas: adição (+), subtração (−), multiplicação (\*), divisão (/) e módulo/resto da divisão inteira (%).

**Nota:** No caso do símbolo − é necessário perceber se este faz parte de uma operação unária ( $x = -a$ ) ou de uma operação binária ( $x = a - b$ ). Em função do tipo de operação, o programa deverá ler 1 ou 2 operandos, respetivamente.

### Exemplo

```
1 Escolha a operacao: *
2 Introduza um operando: 10
3 Introduza outro operando: 4
4 O resultado e igual a: 40
5 Escolha a operacao: -
6 Operador (u)nario ou (b)inario? u
7 Introduza o operando: 7
8 O resultado e igual a: -7
```

## Problema 6

Escreva um programa que leia um ponto (coordenadas  $x, y$ ) e indique em que quadrante se encontra.

### Exemplo

```
1 Indique o ponto (x,y): -4 5
2 O ponto encontra-se no 2o quadrante.
```

## Problema 7

Escreva um programa que leia 3 números e os escreva por ordem ascendente.

### Exemplo

```
1 Insira 3 numeros: 3 1 4
2 1 3 4
3 Insira 3 numeros: 1 4 2
4 1 2 4
```

## 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 é determinar se um ano é bissexto. No entanto, devido à existência de um bug, o programa não desempenha corretamente essa função.

**Nota:** São bissextos todos os anos múltiplos de 4 com a exceção dos anos múltiplos de 100 a menos que também sejam múltiplos de 400 (e.g., o 400 é bissexto enquanto que 100 não é).

**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  int bissextile(int year) {
4      if(year % 4 == 0) {
5          if(year % 100 == 0)
6              if(year % 400 == 0)
7                  return 0;
8              return 1;
9      }
10     return 0;
11 }
12
13 int main() {
14     puts ((!bissextile(100) && !bissextile(5) &&
15         bissextile(4) && bissextile(400)) ? ":" : ": (");
16
17     return 0;
18 }
```

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

**Sugestão:** Experimente adicionar um breakpoint na função “bissextile” e executar o programa interativamente.

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