

Trabalho Prático nº 5: Ciclos

Observações:

- O trabalho deve ser desenvolvido no editor VSCode, com integração ao CodeInsights.
- É importante que identifique corretamente o exercício que está a resolver (exemplo: @aid 5.1) e que utilize as palavras assinaladas a verde como nomes para as variáveis. É igualmente importante que entre `//begin_inputs` e `//end_inputs` sejam apenas declaradas e inicializadas as variáveis que correspondam a inputs do programa.
- Todas as restantes variáveis, constantes e a lógica do programa deverá ser realizada fora deste bloco.
- **Os trabalhos são individuais.** A cópia de trabalhos resultará na reprovação à unidade curricular de todos os alunos envolvidos e outras penalizações previstas no Estatuto Disciplinar dos Estudantes do IPC.

@aid 5.1 Escreva um programa que calcule a soma dos n primeiros números inteiros positivos. O valor de n (variável `num_final`) deve ser introduzido pelo utilizador.

Exemplo de *output* para `num_final=5`:

```
num_final: 5 - soma: 15
```

@aid 5.2 Escreva um programa que leia dois números inteiros (variáveis `inicio` e `fim`) e apresente o produto dos números ímpares compreendidos entre eles (inclusive ambos). Caso o valor de `inicio` seja maior que `fim`, deve ser apresentada a mensagem “intervalo invalido”. Caso não haja números ímpares no intervalo fornecido, deve ser apresentada a mensagem “sem ímpares”.

Exemplo de *output* para `inicio=5` e `fim=10`:

```
de 5 a 10 - ímpares: 5 7 9 produto: 315
```

@aid 5.3 Escreva um programa que leia um número inteiro positivo e inverta os seus dígitos. Utilize a variável `numero`.

Exemplo de *output* para `numero=30427`:

```
numero: 30427 - numero invertido: 72403
```

@aid 5.4 Escreva um programa que leia dois números inteiros `A` e `B` e apresente o máximo divisor comum (MDC) destes números. Utilizar o algoritmo de Euclides, descrito a seguir.

Algoritmo de Euclides: se B é zero, A é o MDC. Caso não seja, calcular o resto da divisão de A por B , copie B para A , copie o resto para B , e repita o processo.

Exemplo de *output* para `A=12` e `B=28`:

```
A: 12 - B: 28 - MDC: 4
```

@aid 5.5 Escreva um programa apresente o calendário de um mês. O utilizador deve introduzir o número de dias do mês (variável **num_dias**) e o dia da semana em que o mês inicia (de 1=domingo a 7=sábado – variável **primeiro**).

Exemplo de *output* para **num_dias=30** e **primeiro=4**:

```
D S T Q Q S S
      1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30
```

@aid 5.6 Um hotel oferece descontos crescentes para clientes, dependendo do número de noites da estadia. Para uma noite apenas não há qualquer desconto, contudo para a segunda noite é aplicado um desconto percentual em relação ao preço normal. Para a terceira noite este desconto percentual é aplicado em dobro, para a quarta noite é triplicado, e assim sucessivamente até um preço mínimo igual a metade do valor sem desconto. Escreva um programa que leia o preço normal (variável real **preco_normal**), a percentagem diária de desconto (variável inteira **desconto_dia**) e o número de noites da estadia (variável **num_noites**), e apresente o preço a pagar por cada noite e o preço total da estadia.

Exemplo de *output* para **preco_normal=65.00**, **desconto_dia=10** e **num_noites=7**:

```
noite 1: 65.00
noite 2: 58.50
noite 3: 52.00
noite 4: 45.50
noite 5: 39.00
noite 6: 32.50
noite 7: 32.50
total: 325.00
```

@aid 5.7 Um número perfeito é um número inteiro positivo para o qual a soma de todos os seus divisores naturais (excluindo ele mesmo) é igual ao próprio número. Por exemplo, o 6 é um número perfeito, pois a soma dos seus divisores, sem o próprio, é: $1+2+3=6$. Escreva um programa que apresente os números perfeitos até um número limite introduzido pelo utilizador (variável **limite**). Caso o limite introduzido seja maior que 10000 (dez mil), o programa deve apresentar a mensagem “limite invalido” e sair.

Exemplo de *output* para **limite=1000**:

```
limite: 1000
numero perfeito: 6 - soma divisores: 6
numero perfeito: 28 - soma divisores: 28
numero perfeito: 496 - soma divisores: 496
fim de procura
```