

ALP

FICHA DE EXERCÍCIOS
ATIVIDADE LETIVA

Algoritmia e Estruturas de Dados

UNIDADE CURRICULAR

Ficha 02 - Estruturas Iterativas

FICHA

■ **Nível 1**

Basic



1. Elabore um programa que simule a função fatorial, isto é, que determine o fatorial de um determinado número.

Exemplo: Fatorial de 5 = $5 * 4 * 3 * 2 * 1 = 120$

Note que $0! = 1$

Nota: **não utilizar a função `math.factorial()`**

O objetivo é desenvolvermos a nossa própria função fatorial.

```
Indique um número: 5
Fatorial de 5 é 120
Press any key to continue . . .
```

```
Indique um número: 0
Fatorial de 0 é 1
Press any key to continue . . . |
```

2. Implemente um programa que peça ao utilizador a indicação de 2 números inteiros (limite inferior e limite superior), calculando em seguida a soma de todos os pares entre esse intervalo (incluindo os limites indicados).

Exemplo:

Limite inferior: 1

Limite superior: 10

Soma dos pares no intervalo = $2+4+6+8+10=30$

```
Indique o limite Inferior:1
Indique o limite superior:10
A soma de todos os pares entre 1 e 10 é 30
Press any key to continue . . . |
```

Intermediate

■ **Nível 2**

3. Jogo Adivinha o número!

Elabore um programa que simule o jogo da adivinha de um número.

O programa deve começar por gerar um número aleatório (entre 1 e 50), permitindo que o jogador tente, iterativamente, adivinhar o número gerado pelo computador.

Tip!

Para gerar um número aleatório recorra à biblioteca `random` (`import random`).

Nesta biblioteca encontra duas funções para gerar números aleatórios:

- `random.randrange(limInf, limSup)`
- `random.randint(limInf, limSup)`

```
import random

num= random.randrange(0,10) # Return random integer in range [a, b[, excludes the end points
num= random.randint(0,10)   # Return random integer in range [a, b], including both end points
```

O jogador dispõe de várias tentativas para adivinhar o número, sendo que após cada tentativa deve surgir uma mensagem do género:

- **"Número é Maior"** - caso o palpite do jogador seja inferior ao número a adivinhar
 - **"Número é Menor"** - caso o palpite do jogador seja superior ao número a adivinhar
 - **"Acertou!!!"** - caso o palpite do jogador coincida com o número a adivinhar.
- Neste caso, o jogo termina com a mensagem **"Parabéns, acertou!!"**

Outras considerações:

- Após 10 tentativas falhadas o jogo deverá terminar, indicando o insucesso do jogador, como uma mensagem do género **"Esgotou as 10 tentativas :("**.
- Quando o jogador acerta no número, o jogo deve indicar o número de tentativas que o jogador necessitou para acertar.

```
JOGO Adivinha o Número

Indique o seu palpite: 25
O número é MAIOR

Indique o seu palpite: 37
O número é MENOR

Indique o seu palpite: 31
Parabéns! Acertou em 3 tentativas
Press any key to continue . . . |
```

4. Faça uma versão 2.0 do jogo anterior em que:

Após a conclusão de um jogo, deve surgir ao utilizador a opção de iniciar um novo jogo: "**Novo jogo (S/N) ?**".

O programa deve comportar-se de acordo com a resposta dada pelo utilizador (**S** ou **N**).

```

                                JOGO Adivinha o Número

Indique o seu palpite: 25
O número é MAIOR

Indique o seu palpite: 37
O número é MAIOR

Indique o seu palpite: 43
Parabéns! Acertou em 3 tentativas
Novo Jogo(S/N)? |

```

Advanced

▪ Nível 3



5. Elabore um programa que leia um número (inteiro e positivo) e indique se ele é **primo** ou não.

Nota: Um número primo é divisível **apenas** por si próprio e por 1.

Números Primos entre 1 e 1000															
Entre 1 e 1000 há 168 números primos, são eles:															
2	3	5	7	11	13	17	19	23	29	31	37	41	43		
47	53	59	61	67	71	73	79	83	89	97	101	103	107		
109	113	127	131	137	139	149	151	157	163	167	173	179	181		
191	193	197	199	211	223	227	229	233	239	241	251	257	263		
269	271	277	281	283	293	307	311	313	317	331	337	347	349		
353	359	367	373	379	383	389	397	401	409	419	421	431	433		
439	443	449	457	461	463	467	479	487	491	499	503	509	521		
523	541	547	557	563	569	571	577	587	593	599	601	607	613		
617	619	631	641	643	647	653	659	661	673	677	683	691	701		
709	719	727	733	739	743	751	757	761	769	773	787	797	809		
811	821	823	827	829	839	853	857	859	863	877	881	883	887		
907	911	919	929	937	941	947	953	967	971	977	983	991	997		

```

Número:21
O numero 21 não é primo
Press any key to continue . . .

```

```

Número:13
O numero 13 é primo
Press any key to continue . . . |

```

6. Elabore um programa que ilustre os primeiros **n** termos da **sequência de Fibonacci**, sendo que o número de termos desejados (**n**) deve ser indicado pelo utilizador.

Na sequência de Fibonacci, cada termo resulta da soma dos dois anteriores.

Fonte: http://pt.wikipedia.org/wiki/N%C3%BAmero_de_Fibonacci

Os **números de Fibonacci** são, portanto, os números que compõem a seguinte [sequência](#) (sequência A000045 na OEIS):
0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, ... ^{[nota 1][2]}.

Exemplo:

```

Nº de termos a imprimir:8

Primeiros 8 termos da sequência de Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13
Press any key to continue . . . |

```

7. Escreva um programa que verifique se um determinado número (inteiro e positivo) é perfeito.

Em Matemática, um número perfeito é um número inteiro para o qual a soma de todos os seus divisores positivos próprios é igual ao próprio número.

Por exemplo, o número 6 é um número perfeito, pois:

6 é divisível por: 1, 2 e 3 $1+2+3 = 6$, logo é um número perfeito

Os quatro primeiros números perfeitos são:

✓ $6 = 1 + 2 + 3$
 ✓ $28 = 1 + 2 + 4 + 7 + 14$
 ✓ $496 = 1 + 2 + 4 + 8 + 16 + 31 + 62 + 124 + 248$
 ✓ $8128 = 1 + 2 + 4 + 8 + 16 + 32 + 64 + 127 + 254 + 508 + 1016 + 2032 + 4064$

C:\WINDOWS\py.exe

```

Indique um número:496
O número 496 é um número perfeito

```

8. Implemente um programa que leia um número (entre 1 e 99) e determine a sua representação em linguagem binária.

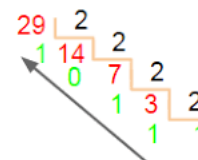
Exemplo:

Número : 12 Resultado: 1 1 0 0
 Número : 29 Resultado: 1 1 1 0 1

```

Número: 29
1 1 1 0 1
Press any key to continue . . .

```



29 Decimal = 11101 Binário

9. Leia um conjunto de n números inteiros (sendo n indicado previamente pelo utilizador). Em seguida determine o segundo maior valor de entre o conjunto de números lido.

Nota: não recorrer a arrays/listas para resolver o exercício!

```
Qts numeros dejesa ler? 6
Número: 10
Número: 12
Número: 14
Número: 16
Número: 8
Número: 9

                Segundo maior valor da lista de números lidos é: 14
Press any key to continue . . . |
```