

Introdução à Programação

Licenciatura em Informática

Ficha nº 3 - Estruturas de Controlo - repetição

Parte I

- 1) Escreva um programa para imprimir os números inteiros entre 1 e 10 na mesma linha, primeiro em ordem crescente e depois em ordem decrescente.
- 2) Escreva um programa para imprimir todos os números inteiros entre dois valores introduzidos pelo utilizador. O programa deverá verificar qual dos dois valores é o maior.
- 3) Escreva um programa que apresente a tabuada dum número inteiro entre 1 e 9 dado pelo utilizador. Se o número estiver fora dessa gama, o programa deverá dar uma mensagem.
- 4) Crie um programa que escreva os números inteiros entre 0 e 100 em intervalos (incremento) dados pelo utilizador. O intervalo deverá ser um número entre 1 e 10. (Por exemplo, com intervalos de 4).
- 5) Escreva um programa que peça ao utilizador um nome e um número inteiro (entre 1 e 20). Deverá mostrar esse nome um número de vezes igual a esse valor inteiro.
- 6) Crie um programa para escrever a série de quadrados entre 1 e um valor inteiro inferior a 100 introduzido pelo utilizador. (1, 4, 9, 25...)
- 7) Escreva um programa que leia 10 números inteiros introduzidos pelo utilizador e indique o máximo, a média, o mínimo e a soma dos valores.
- 8) Escreva um programa em que o utilizador vai introduzindo as idades dos alunos de uma determinada turma até ser introduzido o número -1. No fim deverá indicar o número de alunos e a média de idades. O programa deverá garantir que apenas são introduzidos números positivos (com a exceção do -1 final).
- 9) Escreva um programa em que o utilizador vai introduzindo números positivos até ser introduzido o valor 0 (zero). No fim o programa indicará a percentagem de números pares introduzidos.
- 10) Escreva um programa para ler as notas de n alunos (sendo n introduzido pelo utilizador). As notas deverão estar entre 1 e 5. O programa deverá contar quantos alunos tiveram cada uma das notas possíveis.
- 11) Escreva um programa que leia 10 números inteiros e indique se um número é igual ao anterior. No final deverá indicar quantos números introduzidos são iguais ao anterior.

- 12) [*utilização de flags*]. Escreva um programa que leia *n* números (sendo *n* introduzido pelo utilizador) e indique se os números são todos iguais.
- 13) [*utilização de flags (ou não...)*]. Escreva um programa que leia *n* números (sendo *n* introduzido pelo utilizador) e indique se os números são todos pares, se são todos ímpares ou se há ambos os tipos.
- 14) Escreva um programa que apresente todos os números inteiros entre dois números *float* introduzidos pelo utilizador.
- 15) Escreva um programa que calcule o fatorial de um número.
- 16) Escreva um programa que some os algarismos de um número.
- 17) [*utilização de flags*]. Crie um programa que determine se um número inteiro é primo.
- 18) Escreva um programa em que o utilizador vai introduzindo números inteiros positivos até o número introduzido ser um número primo.
- 19) Escreva um programa que leia 10 números do utilizador e indique, no fim, quantos números são primos, quantos são pares e quantos são divisíveis por 3.
- 20) Escreva um programa em que o utilizador introduza números até introduzir um número par seguido de um número ímpar.
- 21) [*utilização de flags*]. Escreva um programa que leia 10 valores do utilizador e indique no fim se foi introduzido algum número divisível por 7.
- 22) Escreva um programa em que o utilizador introduz números inteiros até introduzir um número em que a soma dos algarismos seja superior a 20.
- 23) Escreva um programa que indique ao utilizador todos os números primos entre dois números inteiros introduzidos pelo utilizador.
- 24) Crie um programa para escrever a série de *Fibonacci* até a um limite pedido pelo utilizador. (1, 2, 3, 5, 8, 13, 21...)
- 25) Crie um programa que imprima um número de 4 dígitos invertido (ex. 4536 -> 6354).
- 26) Escreva um programa que reduza uma fração.
- 27) Aproveite o programa anterior para escrever um programa que realize a soma, subtração, multiplicação e divisão de frações e apresente o resultado na forma reduzida.
- 28) Escreva um programa que, a partir do preço em euros e do dinheiro entregue, apresente o troco usando o menor número possível de moedas.

Parte II

29) **Pedra, Papel, Tesoura:** Escreva um programa que simule o jogo "Pedra, Papel, Tesoura". O utilizador deverá jogar contra o computador, que escolhe aleatoriamente entre as três opções. O jogo termina quando o utilizador decide não continuar.

30) **Adivinhe o Número:** Crie um programa onde o computador escolhe aleatoriamente um número entre 1 e 100. O utilizador deve tentar adivinhar o número. A cada tentativa, o computador indica se o palpite é muito alto, muito baixo ou correto

31) **Números Perfeitos:** Um número é considerado perfeito se a soma dos seus divisores (excluindo ele mesmo) for igual ao próprio número. Escreva um programa que mostre todos os números perfeitos até 1000.

32) **Maior Sequência Crescente:** Escreva um programa que leia uma sequência de números inteiros terminada com o número 0 (zero) e indique a maior sequência crescente encontrada.

33) **Sequência de Collatz:** Dado um número n , a sequência de Collatz é obtida da seguinte forma: se n é par, divide-se por 2; se é ímpar, multiplica-se por 3 e soma-se 1. Repete-se o processo até chegar ao número 1. Escreva um programa que peça um número ao utilizador e mostre a sequência de Collatz correspondente.

34) **Máximo Divisor Comum:** Escreva um programa que leia dois números inteiros e determine o seu Máximo Divisor Comum (MDC).

35) **Números Amigos:** Dois números são amigos se a soma dos divisores de um for igual ao outro número e vice-versa. Peça ao utilizador para introduzir dois números e indique se são amigos.

36) **Sequência de Padovan:** Similar à sequência de Fibonacci, a sequência de Padovan é definida da seguinte forma: $P(n) = P(n-2) + P(n-3)$ com os primeiros valores $P(0) = 1$, $P(1) = 1$, $P(2) = 1$. Escreva um programa que mostre os primeiros n termos da sequência de Padovan, sendo n fornecido pelo utilizador.

37) **Sequência Harmónica:** Escreva um programa que peça ao utilizador um número n e imprima os primeiros n termos da sequência harmónica $(1 + 1/2 + 1/3 + \dots + 1/n)$.

38) **Verificador de Números de Lychrel:** Um número de Lychrel é um número que nunca forma um palíndromo através do processo iterativo de adicionar repetidamente o seu valor ao seu valor invertido. Escreva um programa que peça ao utilizador um número e verifique se ele é um candidato a número de Lychrel em menos de 50 iterações.

39) **Tabuada de Vários Números:** Peça ao utilizador para inserir vários números separados por espaços. Mostre a tabuada de todos esses números.

40) **Sequência Look-and-Say:** Peça ao utilizador um número n e mostre-lhe n termos da sequência Look-and-Say. A sequência começa com "1", e os termos subsequentes são gerados descrevendo o termo anterior (1 -> um 1 -> 11, 11 -> dois 1s -> 21, 21 -> um 2, depois um 1 -> 1211, e assim por diante).

41) **Série Geométrica:** Peça ao utilizador para inserir o primeiro termo, a razão e o número de termos de uma série geométrica. Calcule a soma dos termos dessa série.

42) **Aproximação de π (Pi):** Utilizando a série de Leibniz para π : $\pi = 4 \times (1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + \dots)$, peça ao utilizador o número de termos e calcule uma aproximação para π .

43) **Jogo das Varetas:** Simule o jogo das varetas, onde há inicialmente 21 varetas e os jogadores, alternadamente (utilizador e computador), escolhem retirar entre 1 e 4 varetas. O jogador que retirar a última vareta perde.

44) **Números de Armstrong:** Um número é considerado de Armstrong se a soma dos cubos dos seus dígitos for igual ao próprio número (para números de 3 dígitos). Peça ao utilizador um intervalo e mostre todos os números de Armstrong nesse intervalo.

45) **Média Ponderada:** Peça ao utilizador para introduzir várias notas e os respetivos pesos. Calcule e apresente a média ponderada dessas notas.