

AVALIAÇÃO

A avaliação da unidade curricular (UC):

- Teste ou exame escrito (20% nota final) e implementação e apresentação dos trabalhos práticos (80% nota final);
- Na componente “Teste ou exame” a nota mínima para obter a aprovação da UC é de 7 valores (0-20);
- Na componente “implementação e apresentação dos trabalhos práticos” a nota mínima para obter a aprovação da UC é de 7 valores (0-20);
- Os trabalhos podem ser realizados individualmente ou em grupos até 4 elementos, mas a apresentação é individual;
- Trabalhos/exercícios apresentados durante a aula(s) onde são propostos e previstos desenvolver (ver em baixo), terão a cotação de 0-20 valores;
- O último dia para apresentar os trabalhos/exercícios é o dia do exame de recurso.

Trabalhos práticos (20val) =

Aula	TI	Aula	STI	Aula	TI	Aula	STI
1	1,00	1	1,00	7	2,50	6	2,75
2	1,25	2	1,50	8	2,25	7,8	2,50
3	1,50	3	2,00	9,10	2,25	9-11	2,50
4,5	2,50	4	2,50	11-13	2,50	12-14	2,75
6	2,00	5	2,50	14-20	2,25	--	0,00

AULA TI 1 & STI 1

Exercício 1

Converta para decimal os seguintes números binários:

101 =

1101 =

120 =

Converta para binário os seguintes números decimais:

120 =

255 =

1 =

Adicione os seguintes números em binário:

1001 + 001 =

Quantos bits e quantos Bytes tem o seguinte número:

0000100111011101 =

Exercício 2

Faça o algoritmo: Calcule a distância entre dois pontos, sendo que cada ponto é definido pelas coordenadas (x,y), i.e., ponto 1 (x1,y1) e ponto 2 (x2,y2), utilizando a seguinte fórmula: $\text{distancia} = \text{raiz}((x2 - x1)^2 + (y2 - y1)^2)$.

Exercício 3

Faça o algoritmo: Leia as notas de um aluno às disciplinas de Matemática, Português, Inglês e Geografia e calcule a média. Em função da média mostra uma mensagem com o conteúdo "Aprovado" ou "Reprovado". Consideram-se notas positivas as notas iguais ou superiores a 9,5.

Exercício 4

Faça o algoritmo: Calcular o fatorial de um número, i.e.,

$$\text{factorial}(n) = \begin{cases} n = 0 & \rightarrow 1 \\ n \geq 1 & \rightarrow n * \text{factorial}(n - 1) \end{cases}.$$

Exercício 5

Faça o algoritmo: Determinar se um número é primo (um número é primo se for apenas divisível por si próprio e pela unidade).

Exercício 6

Faça o algoritmo: Determinar o aluno melhor classificado e a média das notas de uma turma. Permitir ler as notas de programação obtidas pelos elementos de uma turma. O programa termina quando for introduzido no nome do aluno "STOP". No final deve ser mostrado o nome do aluno melhor classificado e a média de notas.

Exercício 7

Faça o algoritmo: Escreva uma função que recebe um número inteiro e devolve o maior algarismo contido nesse número. Ex. 12634 -> devolve o "6"

Exercício 8

Faça o algoritmo: Escreva uma função que num vetor faça a pesquisa sequencial de um valor “valor”.

AULA TI & STI 2

Exercício 1

Implemente um programa que escreva como output “Olá mundo” 10 vezes.

Exercício 2

Escreva o algoritmo e o programa para calcular a fórmula resolvente de equações de 2º grau,

$$s = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Os valores a, b, e c, devem ser gerados aleatoriamente. O valor de a varia entre [0, 10], e os valores b e c variam entre [5, 20]. Se $(b^2 - 4ac) < 0$, o output deve ser “não existem raízes reais”.

Exercício 3

Implemente um programa para trocar euros em dólares considerando a taxa de conversão 1,17.

Exercício 4

Implemente um programa para determinar perímetro e área de circunferência.

Exercício 5

Implemente um programa para que verifique se o aluno é aprovado (se tem nota maior ou igual a 9,5).

Exercício 6

Implemente um programa para calcular a área e verifica se é um quadrado ou um retângulo.

Exercício 7

Implemente um programa para modelar o funcionamento de uma máquina. Sendo que quando da inserção das letras: 'L', 'D' e 'F', são apresentadas, respetivamente, as mensagens: “Ligar”, “Desligar” e “Furar” (use if...else).

Exercício 8

Implemente um programa para determinar o máximo de 3 valores. Ex. 123, 23 e 4 -> retorna máximo é “123”.

Exercício 9

Implemente um programa para calcular:

- (a) a média das idades e das alturas de todos os alunos da turma;*
- (b) qual a altura do aluno mais alto*
- (c) qual a idade do aluno mais novo*

AULA TI 3 & STI 3

Exercício 1

Implemente um programa para determinar se um número é primo (um número é primo se for apenas divisível por si próprio e pela unidade).

Exercício 2

Implemente um programa para calcular a distância entre dois pontos, sendo que cada ponto é definido pelas coordenadas (x,y), utilizando a seguinte fórmula: $\text{distância} = \text{raiz}((x_2 - x_1)^2 + (y_2 - y_1)^2)$.

Exercício 3

Implemente um programa para leia as notas de um aluno às disciplinas de Matemática, Português, Inglês e Geografia e calcule a média. Em função da média mostra uma mensagem com o conteúdo "Aprovado" ou "Reprovado". Consideram-se notas positivas as notas iguais ou superiores a 9,5. Garanta que o valor da nota introduzida está situado entre 0 e 20.

Exercício 4

Implemente um programa para fazer a tabuada de um número lido.

Exercício 5

Implemente um programa para calcular a somatório dos números existentes num intervalo definido por limites inferior e superior.

Exercício 6

Implemente um programa para calcular o fatorial de um número.

Exercício 7

Implemente um programa para determinar nome e idade da pessoa mais nova de um grupo. Ler o número do Cartão de Cidadão e a idade de uma série de pessoas, e o programa deve terminar quando for introduzido a idade da pessoa = 999. No final deve ser mostrado o nome e idade da pessoa mais nova.

Exercício 8

Escreva o algoritmo e o programa para fazer a factoração de um número inteiro (fatorar é o mesmo que decompor o número em fatores primos, isto é, escrever um número através da multiplicação de números primos. Na factoração utilizamos os números primos obedecendo a uma ordem crescente de acordo com as regras de divisibilidade em razão do termo a ser fatorado. Números primos são aqueles que podem ser divididos somente por um e por ele mesmo. Observe a decomposição em fatores primos dos números a seguir:

$$24 = 2 \times 2 \times 2 \times 3$$

$$10 = 2 \times 5$$

$$52 = 2 \times 2 \times 13$$

$$112 = 2 \times 2 \times 2 \times 2 \times 7$$

$$600 = 2 \times 2 \times 2 \times 3 \times 5 \times 5$$

Forma prática de factoração: O número a ser fatorado deverá ocupar a coluna da esquerda e a coluna da direita será preenchida com os fatores primos. Ao dividir o número pelo algarismo primo os resultados deverão ser colocados na coluna da direita. As divisões deverão ser efetuadas no intuito de simplificar ao máximo o número, isto é reduzi-lo ao número 1).

Exercício 9

Escreva o algoritmo e o programa para todos os estudantes do TeSP do ISE-DEE por turma, ler (não use vetores e/ou matrizes e/ou strings):

- (a) a idade;*
- (b) o género (masculino – 1; feminino – 0);*
- (c) nota de entrada.*

Apresente:

- (i) qual a turma com o aluno com o aluno mais novo*
- (ii) qual a turma com o aluno mais velho*
- (iii) quantas raparigas existem nos TeSP-DEE*
- (iv) qual a turma com o aluno com a nota de entrada mais alta*
- (v) qual a nota de entrada mais alta e o género da pessoa que teve essa nota*

AULA TI 4, 5 & STI 4

Exercício 1

Implemente um programa que permita a leitura de 10 valores (inteiros), guardando-os numa lista, imprima a diferença entre o maior e o menor valor do vetor, bem como o número de valores pares e ímpares do vetor.

Exercício 2

Implemente um programa que ordene uma lista de N elementos.

Exercício 3

Implemente um programa que pesquise o número “valor” numa lista de N elementos.

Exercício 4

*Implemente o algoritmo e implemente o programa do **jogo galo** (jogo da velha ou três em linha) (https://pt.wikipedia.org/wiki/Jogo_da_velha).*

Exercício 5

*Implemente o algoritmo e implemente o programa do **jogo 4 em linha** (<https://brainking.com/pt/GameRules?tp=13>).*

AULA TI 6 & STI 5

Exercício 1

Implemente um programa que tenha uma função de forma a calcular a potência. Leia o número e o expoente.

Exercício 2

Implemente um programa que tenha uma função que recebe um número inteiro e devolve o maior algarismo contido nesse número.

Exercício 3

Implemente um programa que tenha uma função que receba um número inteiro e devolve o menor algarismo contido nesse número.

Exercício 4

Implemente um programa que permita: a) Uma função que faça a leitura de 10 valores (inteiros), guardando-os num vetor; b) Uma função que retorne a diferença entre o maior e o menor valor do vetor; c) Uma função que devolva o número de valores pares e ímpares do vetor.

Exercício 5

Implemente um programa que tenha uma função para ordenar um vetor, retornando esse vetor ordenado.

Exercício 6

Implemente um programa que tenha uma função para procurar um número num vetor, retornando “true” caso detete o número, “false” caso contrário.

Exercício 7

Implemente um programa que tenha uma função que leia o nome do utilizador (ex. João Miguel Rodrigues) e coloque o apelido em primeiro lugar seguida de uma virgula (ex. Rodrigues, João Miguel)

Exercício 8

Implemente um programa que tenha funções que leiam o nome das disciplinas (UC) do presente semestre e a nota esperada para cada uma. Retire as iniciais (sigla) e de cada UC (ex. Programação – P; Programação Orientada Objetos – POO), e que coloque no ecrã a sigla – nota.

AULA TI 7 & STI 6

Exercício

Faça um programa que permite selecionar e jogar os seguintes jogos (use funções):

(a) Jogo do galo;

(b) Jogo 4 em linha;

(c) Jogo da gloria (ex. <https://bit.ly/2Ucrusu>; <https://bit.ly/3xBwi8j>; <https://bit.ly/3xxsjcU>)

AULA TI 8 & STI 7, 8

Exercício

Faça o algoritmo e implemente o programa do Jogo da força (https://pt.wikipedia.org/wiki/Jogo_da_forca)

AULA TI 9, 10 & STI 9-11

Exercício

(a) Faça um programa que permite selecionar e jogar os seguintes jogos (use funções): Jogo do galo;

Jogo 4 em linha;

Jogo da glória;

Jogo da força;

Jogo campo de minas (https://pt.wikipedia.org/wiki/Campo_minado).

- (b) *Cada vez que pretende terminar sair de qualquer dos jogos sem o ter terminado, deve guardar o status do jogo (use ficheiros). O mesmo acontece quando pretende terminar o programa.*

Cada vez que entra no jogo deve ser perguntado se pretende continuar o jogo anterior (caso tenha saído a meio) ou se pretende iniciar um novo jogo.

. Deve guardar:

- o nome dos jogadores que estão a jogar;

- a pontuação por jogo e total;

Quando iniciar o programa deve ser perguntado se quer começar nova “sessão de jogos” ou se pretende continuar uma que tenha guardado.

- (c) *Use programação modular. Cada um dos jogos referidos em (a) deve estar num “ficheiro separado”.*

AULA TI 11-13 & STI 12-14

Exercício

*Ao exercício anterior adicione o jogo da **batalha naval***

([https://pt.wikipedia.org/wiki/Batalha_naval_\(jogo\)](https://pt.wikipedia.org/wiki/Batalha_naval_(jogo))), com 2 vertentes:

- (a) *Jogador vs Computador: o computador deve escolher a posição onde são colocados os barcos, o computador escolhe as posições dos tiros aleatoriamente (opção: ou com algumas regras).*
- (b) *Jogador1 vs Jogador2: cada jogador escolhe a posição dos seus barcos.*

AULA TI 14 – 20 (apenas para TI)

Exercício

Implemente um programa para criar um bundle de jogos tradicionais, nomeadamente:

Jogo do galo;

Jogo 4 em linha;

Jogo da glória;

Jogo da força;

Jogo campo de minas;

Jogo da batalha naval;

***Jogo moinho** (ou trilha)* (apenas para CTeSP-TI).*

Considere a vitória num jogo 3 pontos, empate 1 ponto derrota 0 pontos.

- (a) *Todos os jogos devem poder ser jogados entre 2 jogadores e entre o computador e 1 jogador;*
- (b) *Em qualquer momento pode ser gravado o estado dos jogos para “disco”;*
- (c) *Em qualquer momento podem ser lidos os jogos gravados em “disco”;*
- (d) *Em qualquer momento pode se obter o nome dos jogadores que estão a jogar e a sua pontuação (essa pontuação é cumulativa de todos os jogos jogados na “sessão”);*

* Ex. [https://pt.wikipedia.org/wiki/Trilha_\(jogo\)](https://pt.wikipedia.org/wiki/Trilha_(jogo)); <https://bit.ly/3xxsjcU>