|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | |  | |  | |  |  | |  |  |  | |  |
| **AVALIAÇÃO** | | | **QUESTIONÁRIO:**  **FICHA:  TRABALHO:  FORMATIVO:  SUMATIVO:** | | | | | | | | | | | |  |
|  |  |  | |  | |  | |  |  | |  |  |  | |  |
| **Curso:** | EFA T Programador/a | | | | **Turma:** | | 02-20 | | | **Data:** | 2021-03-08 | **Nº Pág.:** | | 1 |  |
|  |  |  | |  | |  | |  |  | |  |  |  | |  |
| **UFCD:** | 11, cód.: 0804, Algoritmos | | | | **Formador:** | | | Rui Boticas | | | | **Classificação:** | | 20 |  |
|  |  |  | |  | |  | |  |  | |  |  |  | |  |

|  |  |
| --- | --- |
| **Objetivo 1** | **Desenvolver algoritmos** |
|  | * Contruir algoritmos com representação genérica para qualquer tipo de problema. * Contruir algoritmos com representação especifica. * Contruir algoritmos com representação genérica e aplicáveis a qualquer tipo de problema de génese igual ou similar. * Contruir algoritmos com representação genérica para problemas informáticos padrão. |

|  |  |
| --- | --- |
| **Objetivo 2** | **Reconhecer algoritmos** |
|  | * Reconhecer algoritmos padrão para tipos de problemas já solucionados. |

|  |  |
| --- | --- |
| **Objetivo 3** | **Representação gráfica de algoritmos** |
|  | * Representação de algoritmos através de fluxogramas. |

|  |  |
| --- | --- |
| **Duração** | **Início: 2021-03-10 – 09H00** |
|  | * **Tempo**: 180 minutos; |

|  |  |
| --- | --- |
| **Entrega** | **2021-03-10 – 12H00.** |
|  | * Tarefa Microsoft TEAMS. |

|  |  |
| --- | --- |
| **Formato** | **Por TEAMS.** |
|  | * Tarefa |

***Enunciado***

|  |  |
| --- | --- |
|  | **Grupo 1 - Algoritmia Formal (8 pontos)** |

* 1. Crie um algoritmo que peça dois valores ao utilizador e se ambos forem negativos, é-lhe mostrado a soma entre eles, caso contrário, é mostrada uma mensagem de erro.
  2. Crie um algoritmo para um programa que peça dois valores ao utilizador e lhe mostre qual foi o menor valor introduzido.
  3. Crie um algoritmo que peça doze valores ao utilizador e diga qual foi o menor de todos os valores introduzidos.

|  |  |
| --- | --- |
|  | **Grupo 2 - Algoritmia Formal (6 pontos)** |

Considere os seguintes problemas e crie o algoritmo para os mesmos:

* 1. É necessário construir uma calculadora que mostre ao utilizador o cubo de números (número x número x número) ímpares, maiores que 5 e menores que 100. Se o número introduzido estiver fora daquela desse gama, o utilizador deverá ser avisado de como deverá proceder para a calculadora funcionar corretamente.
  2. Dado um array de números introduzidos do teclado, é necessário indicar em que posição está o primeiro número acima de um valor introduzido pelo utilizador.
  3. É necessário saber qual é a letra a seguir à letra que o utilizador colocar no teclado (atenção isto apenas pode funcionar se o utilizador introduzir uma letra válida).

|  |  |
| --- | --- |
|  | **Grupo 3 - Algoritmia Formal (3 pontos)** |

Considere o seguinte algoritmo:

INICIO

VAR lista\_numeros <- ler\_teclado()

VAR i <- 0, resultado <- verdade

ENQUANTO(i < tamanho(lista\_numeros) - 1)

SE(lista\_numeros[i] > lista\_numeros[i+1])

Resultado <- falso

fSE

i <- i + 1

fENQ

FIM

* 1. Indique o que o programa faz em português (atenção!! o que se pretende com este exercício, é que descreva o sentido global do programa e **NÃO** que faça a descrição de cada uma das instruções);
  2. Desenhe o fluxograma para o algoritmo dado (seja fiel ao algoritmo);

|  |  |
| --- | --- |
|  | **Algoritmia Gráfica (Fluxograma) (3 pontos)** |

* 1. Crie um fluxograma para a alínea 2.1. Caso não consiga fazer a questão 2.1, em alternativa poderá fazer para a 1.3.