

Resolução**Algoritmos Extra**

Escreva um algoritmo, em pseudocódigo, para resolver cada um dos seguintes problemas.

1. *Dado um valor de Temperatura em graus Celsius, Farenheit ou Kelvin, converter esse valor para as restantes unidades e apresentar os valores. (Nota: $^{\circ}\text{C} = (^{\circ}\text{F} - 32) * 5/9$; $^{\circ}\text{K} = ^{\circ}\text{C} + 273.15$)*

nome: converterTemperaturas { }

variáveis de entrada:

variáveis de saída:

variáveis: valIni, valFim1, valFim2: número real

escalaIni: carácter

INICIO

escrever("introduza o valor inicial")	}	ler(valIni)
receber(valIni)		
escrever("introduza a escala inicial")	}	ler(escalaIni)
receber(escalaIni)		

SE (escalaIni == 'C') ENTÃO

INICIO

valFim1 = valIni * 9/5 + 32 (C. para F.)
valFim2 = valIni + 273.15 (C. para K.)
escrever(valFim1, valFim2) (apresentar os resultados)

FIM

SE (escalaIni == 'F') ENTÃO

INICIO

valFim1 = (valIni - 32) * 9/5 (F. para C.)
valFim2 = valFim1 + 273.15 (F. para K. utilizando o valor de C.)
escrever(valFim1, valFim2) (apresentar os resultados)

FIM

SE (escalaIni == 'K') ENTÃO

INICIO

valFim1 = valIni - 273.15 (K. para C.)
valFim2 = valFim1 * 9/5 + 32 (K. para F. utilizando o valor de C.)
escrever(valFim1, valFim2) (apresentar os resultados)

FIM

FIM

Resolução**Algoritmos Extra****2. Verificar e indicar se um número fornecido é positivo.**

nome: ePositivo { }

variáveis de entrada:

variáveis de saída:

variáveis: num: número inteiro

INICIO

ler(num)

SE (num > 0) ENTÃO

escrever("o num é positivo")

SENÃO

escrever("não positivo")

FIM

Outra versão em que o número não é introduzido pelo teclado, mas surge como variável de entrada inicial. A indicação final também não é apresentada no ecrã, mas passada para o exterior como variável de saída:

nome: ePositivoV2 {num, indicador}

variáveis de entrada: num: inteiro

variáveis de saída: indicador: inteiro

INICIO

SE (num > 0) ENTÃO

indicador = 1 (optou-se pelo 1 por uma questão de convenção)

SENÃO

indicador = 0 (optou-se pelo 0 por uma questão de convenção)

FIM

3. Verificar e indicar se um número inteiro é par ou ímpar.

nome: parOuImpar { }

variáveis de entrada:

variáveis de saída:

variáveis: num: número inteiro

INICIO

ler(num)

SE (resto(num/2) == 0) ENTÃO

escrever("o num é par")

SENÃO

escrever("o num é ímpar")

FIM

Resolução**Algoritmos Extra***4. Determinar e apresentar o maior de dois números introduzidos pelo utilizador.*

nome: maiorNum { }

variáveis de entrada:

variáveis de saída:

variáveis: num1, num2: número inteiro

INICIO

ler(num1)

ler(num2)

SE (num1 > num2) ENTÃO

escrever("num1 > num2")

SENÃO SE (num2 > num1) ENTÃO

escrever("num2 > num1")

SENÃO

escrever("são iguais")

FIM

5. Verificar e indicar se um número fornecido se encontra entre 1 e 10.

nome: entreLimites1 { }

variáveis de entrada:

variáveis de saída:

variáveis: num: número inteiro

INICIO

ler(num)

SE (num >= 1 E num <= 10) ENTÃO (em vez do E podes utilizar: &&, AND, ^)

escrever("está ok")

SENÃO

escrever("KO")

FIM

6. Receber um número inteiro que esteja obrigatoriamente entre 1 e 10.

nome: entreLimites2 { }

variáveis de entrada:

variáveis de saída:

variáveis: num: número inteiro

INICIO

REPETIR

ler(num)

ENQUANTO (num < 1 OU num > 10) (em vez do OU pode ser ||, OR, entre outros.)

Resolução**Algoritmos Extra**

escrever("ok – até que enfim!")

FIM

7. *Descobrir um número entre 0 e 100, pensado pelo utilizador. Após ter acertado deve informar o utilizador do número de tentativas efectuadas.*

8. *Determinar se um dado número é primo.*

nome: ePrimo {num, indicador}

variáveis de entrada: num: inteiro

variáveis de saída: indicador: inteiro

variáveis: div: inteiro

INICIO

div = 2

REPETIR ENQUANTO (resto(num/div) != 0) (em vez de != pode ser <>, dif., entre outros.)

div = div + 1

SE (div == num)

indicador = 1

SENÃO

indicador = 0

FIM

9. *Somar os primeiros 20 números inteiros (positivos) (são os naturais).*

nome: somar20 { }

variáveis de entrada:

variáveis de saída:

variáveis: num, soma: número inteiro

INICIO

num = 1

soma = 0

REPETIR ENQUANTO (num <= 20)

INICIO

soma = soma + num

num = num + 1

FIM

escrever(soma)

FIM

Resolução**Algoritmos Extra**

10. Somar um conjunto de 20 números inteiros fornecidos pelo utilizador.

nome: somar20_user { }

variáveis de entrada:

variáveis de saída:

variáveis: num, soma, conta: número inteiro

INICIO

 conta = 1

 soma = 0

 REPETIR ENQUANTO (conta <= 20)

 INICIO

 ler(num)

 soma = soma + num

 conta = conta + 1

 FIM

 escrever(soma)

FIM

11. Determinar o valor da soma dos primeiros N números primos.

Este algoritmo utiliza o algoritmo 8.

nome: somarNPrimos { }

variáveis de entrada:

variáveis de saída:

variáveis: n, num, conta, soma: número inteiro

INICIO

 ler(n)

 num = 2 (iniciar pela análise ao 2)

 soma = 0

 conta = 0

 REPETIR ENQUANTO (conta < n)

 INICIO

 SE (ePrimo(num) == 1) (ao utilizar o ePrimo, se este indicar 1 então o num é primo)

 INICIO

 soma = soma + num

 conta = conta + 1

 FIM

 num = num + 1

 FIM

Resolução**Algoritmos Extra**

escrever(soma)

FIM

12. Determinar quanto números primos existem num determinado intervalo recebido do utilizador.

nome: contarPrimos { }

variáveis de entrada:

variáveis de saída:

variáveis: limInf, limSup, num, conta: número inteiro

INICIO

ler(limInf)

ler(limSup)

num = limInf

conta = 0

REPETIR ENQUANTO (num <= limSup)

INICIO

SE (ePrimo(num) == 1)

conta = conta + 1

num = num + 1

FIM

escrever(conta) (apresentar a contagem ao utilizador)

FIM