//Exercícios resolvidos por Guilherme Castro —- BSI-125 1/2025

1. Faça um programa para calcular e exibir a soma dos cem primeiros números inteiros (1+2+3+4+5+...+99+100).

algoritmo

declare

n1, n2, n3, resultado numerico

n1 <- 1

n2 <- 100

repita

n3 <- n1 + n2

resultado <- resultado + n3

n1 <- n1 + 1

n2 <- n2 - 1

ate n1 = 51

escreva resultado

fim\_algoritmo

2. Faça um programa para ler 10 números e calcular e exibir sua média aritmética.

algoritmo

declare

n, n2, c, media numerico

c <- 1

repita

escreva "Insira o ", c,"º número:"

leia n

n2 <- n2 + n

c <- c + 1

ate c > 10

escreva "A média dos números informados é de: ", n2/10

fim\_algoritmo

3. Faça um programa para ler um número inteiro N qualquer maior ou igual 1 e menor ou igual a 10 e, calcular e exibir os resultados da tabuada de N.

Exemplo: N = 6 Tabuada:

6x1=6

6x2=12

6x3=18…

6x10=60.

O valor de N deve ser checado quanto a sua validade. Caso o usuário informe umvalor fora da faixa, o programa deve informa por meio de uma mensagem que o valor digitado não é válido.

algoritmo

declare

numero, numero2, resultado numerico

escreva "Insira um número maior que 1 e menor ou igual a 10:"

leia numero

se numero < 1 ou numero > 10 entao

inicio

repita

escreva "Número inválido, insira um novo valor:"

leia numero

ate numero >= 1 e numero <= 10

fim

repita

numero2 <- numero2 + 1

resultado <- numero\*numero2

escreva numero, " X ", numero2, " = ", resultado

ate numero2 = 10

fim\_algoritmo

4. Faça um programa para ler um número inteiro N qualquer maior ou igual 1 e menor ou igual a 50 e, calcular e exibir o valor obtido pela multiplicação sucessiva de N por 2, enquanto o produto for menor

que 250 (Nx2;Nx2x2;Nx2x2x2; etc.). O valor de N deve ser checado quanto a sua validade. Caso o usuário informe um valor fora da faixa, o programa deve informa por meio de uma mensagem que o valor digitado não é válido.

algoritmo

declare

nm, produto numerico

escreva "Digite um número entre 1 e 50:"

leia nm

se nm < 1 ou nm > 50 entao

inicio

repita

escreva "Número inválido, insira outro:"

leia nm

ate nm > 1 e nm < 50

fim

enquanto produto < 250 faca

inicio

produto <- nm \* 2

se produto < 250 entao

escreva nm, " X 2 =", produto

nm <- produto

fim

fim\_algoritmo

5. Faça um programa para exibir todos os números inteiros ímpares situados na faixa de 0 a 100. Para verificar se o número é ímpar, o programa deverá checar se o resto da divisão do número por 2 é igual a

zero.

ALGORITMO

DECLARE

numero NUMERICO

repita

numero <- numero + 1

se resto(numero, 2) = 1 entao

escreva numero

ate numero = 100

fim\_algoritmo

6. Faça um programa para exibir todos os números inteiros positivos divisíveis por 7 e menores que 200.

Para verificar se o número é divisível por 7, o programa deverá checar se o resto da divisão do número por 7 é igual a zero.

ALGORITMO

DECLARE

numero NUMERICO

REPITA

numero <- numero + 1

SE resto(numero, 7) = 0 ENTAO

ESCREVA numero

ATE numero = 200

fim\_algoritmo

7. Faça um programa para exibir todos os números inteiros positivos e menores que 500 que quando divididos por 9 deixam resto 4.

algoritmo

DECLARE

numero NUMERICO

numero <- 8

repita

numero <- numero + 1

SE resto(numero, 9) = 4 entao

escreva numero

ate numero = 500

fim\_algoritmo

8. Faça um programa para exibir os quadrados dos números inteiros de 15 a 200.

algoritmo

declare

numero numerico

numero <- 15

repita

escreva numero,"² = ",potencia(numero, 2)

numero <- numero + 1

ate numero = 200

fim\_algoritmo

9. Faça um programa para gerar e imprimir os 25 primeiros termos da seguinte série : 1,3,5,7,9,11,...

algoritmo

declare

numero numerico

repita

numero <- numero + 1

se resto(numero, 2) <> 0 entao

escreva numero

ate numero = 25

fim\_algoritmo

10. Sendo:

Fazer um programa para solicitar N e gerar

o número H.

algoritmo

declare

N, H numerico

escreva "Insira um valor para ''N''"

leia N

repita

H <- H + (1/N)

N <- N - 1

ate N = 0

escreva "H = ", H

fim\_algoritmo

11. A sequência abaixo é conhecida como série de Fibonacci. Faça um programa para escrever esta série até o seu trigésimo termo. Série de Fibonacci : 1,1,2,3,5,8,13,21,34,55,...

algoritmo

declare

n1, n2, acumulador, contador numerico

n1 <- 1

n2 <- 1

contador <- 1

escreva contador, "º: ", n1

contador <- 2

escreva contador, "º: ", n2

repita

contador <- contador + 1

acumulador <- n1 + n2

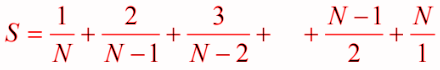
n2 <- n1

n1 <- acumulador

escreva contador, "º: ", acumulador

ate contador = 30

fim\_algoritmo

12. Fazer um programa para solicitar N e calcular o valor de S, dado por:

algoritmo

declare

S, N, Na,Nb numerico

escreva "Insira o valor de ''N'':"

LEIA N

Na <- 1

Nb <- N

repita

S <- S + Na/Nb

Na <- Na + 1

Nb <- Nb - 1

escreva S

ate Na > N

fim\_algoritmo

13. Dado um país A, com 5.000.000 de habitantes e uma taxa de natalidade de 3% ao ano, e um país B com 7.000.000 e uma taxa de natalidade de 2% ao ano, construa um programa que calcula quantos anos serão necessários para que a população do país A ultrapasse a população do país B.

algoritmo

declare

A, B, anos numerico

A <- 5000000

B <- 7000000

enquanto A < B faca

inicio

anos <- anos + 1

A <- A + (3/100\*A)

B <- B + (2/100\*B)

escreva "Ano ", anos

escreva "País A: ", arredonda(A)

escreva "País B: ", arredonda(B)

fim

escreva anos, " anos para a população de A passar B."

fim\_algoritmo

14. Um rei requisitou os serviços de um sábio e disse-lhe que pagaria qualquer preço. O sábio, necessitando de alimento para seu povo, solicitou ao rei que o pagamento por seus serviços fosse feito em grãos de trigo e respeitando a seguinte regra: para contar a quantidade de grãos usariam um tabuleiro de xadrez; o rei dar-lhe-ia um grão pela primeira casa do tabuleiro, dois grãos pela segunda casa, quatro pelo terceira, oito pela quarta, e, assim dobrando sucessivamente, até a sexagésima quarta e última casa do tabuleiro. O

rei achou barato e ordenou que o pagamento fosse efetuado, sem se dar conta de que seria impossível realizar o pagamento. Faça um programa para calcular o número de grãos que o rei deve ao sábio.

algoritmo

declare

grao, i numerico

repita

i <- i + 1

grao <- potencia(2, i) - 1

escreva i, "º quadro: ", grao

ate i = 64

fim\_algoritmo

15. Suponha que uma dada loja venda as suas mercadorias por um real ou menos. Assuma ainda que todos os clientes pagam pelas mercadorias compradas com uma nota de um real. Preparar um programa que leia o

preço de um produto e calcule o número de cada tipo de moeda a ser dado de troco, de modo que o menor número de moedas é retornado ao cliente. Por exemplo, se o valor de uma compra é 63 centavos de real,

o troco terá 1 moeda de 25, 1 moeda de 10 e 2 moedas de 1 centavo. OBS : moedas válidas => 1, 5, 10, 25 e 50 centavos.

algoritmo

declare

valor, troco, moeda numerico

escreva "Digite um número menor que 1"

leia valor

enquanto valor > 1 faca

inicio

escreva "Digite um valor menor que 1."

leia valor

fim

valor <- valor \* 100

troco <- 100 - valor

escreva "Troco: R$", troco/100

enquanto troco >= 50 faca

inicio

moeda <- moeda + 1

troco <- troco - 50

fim

se moeda <> 0 entao

escreva "Moedas de 50: ", moeda

moeda <- 0

enquanto troco >= 25 faca

inicio

moeda <- moeda + 1

troco <- troco - 25

fim

se moeda <> 0 entao

escreva "Moedas de 25: ", moeda

moeda <- 0

enquanto troco >= 10 faca

inicio

moeda <- moeda + 1

troco <- troco - 10

fim

se moeda <> 0 entao

escreva "Moedas de 10: ", moeda

moeda <- 0

enquanto troco >= 5 faca

inicio

moeda <- moeda + 1

troco <- troco - 5

fim

se moeda <> 0 entao

escreva "Moedas de 5: ", moeda

moeda <- 0

enquanto troco >= 1 faca

inicio

moeda <- moeda + 1

troco <- troco - 1

fim

se moeda <> 0 entao

escreva "Moedas de 1: ", moeda

fim\_algoritmo

16. O máximo divisor comum de dois inteiros é o maior número que divide ambos sem deixar resto. Escreva um programa que lê dois inteiros e calcula o seu máximo divisor comum.

algoritmo

declare

n1, n2, a, b, x numerico

escreva "Insira um número:"

leia n1

escreva "Insira um número:"

leia n2

se n1 > n2 entao inicio

a <- n1

b <- n2

fim

senao inicio

a <- n2

b <- n1

fim

enquanto b <> 0 faca

inicio

x <- resto(a, b)

a <- b

b <- x

fim

escreva "Máximo Divisor Comum de ", n1, " e ", n2, " :", a

fim\_algoritmo