# codigo em python para resolver a questão proposta

def calculaFrequencia(presencaAluno,horasAula: float):

return ((presencaAluno\*100)/horasAula)

if ( (calculaFrequencia(75.01,10)) < 75.0):

print("Estudante reprovado pela frequencia em sala de aula")

else:

print("Estudante Aprovado pela frequencia em sala de aula")

# keywords para a “MLP”(Minha Linguagem de Programação)

\\ as duas barras laterais indicam um comentário a ser feito

{\ sintaxe para comentário em bloco \}

Ex.:

{\

Este eh um

Bloco comentado

Tudo aqui o que estiver entre “{\”

E o “\}” será ignorado

\}

/ \\ a barra lateral e utilizada para imprimir o valor da variável em uma expressão imprime(), segue exemplo

Ex.:

#declare valor: Integer;

valor= 10;

imprime(“O valor da variável eh /valor”,&valor);

\\ para imprimir o valor da variável com formatação, ou seja, entre no texto utiliza-se a seguinte forma: imprime(“Texto /elemento1”,&variavel1), em caso de múltiplas variáveis vide o próximo exemplo:

Ex.:

#declare valor1, valor2, valor3: Integer;

#declare car: String;

Imprime(“Este e meu texto formatado valor da variável 1: /elemento1 e em seguida o valor da variável 2: /elemento2 e o valor da variável 3 /elemento3 e agora o valor da variável que contém uma string /elemento4”,&valor1,&valor2,valor3,$car);

mathSigma() \\ soma todos os valores passados como parâmetro

mathDiv() \\ divisão dos números passados como parâmetro,

ex.: mathDiv(6,3) o resultado será 2 OBS.: sempre tomando 2 a 2

mathEXP() \\ eleva o primeiro número passado como parâmetro ao segundo número passado como parâmetro

imprime() \\ imprime os valores passados como parâmetro no terminal

recebe() \\ utilizado para inserir um valor digitado no terminal

#declare \\ utilizado para declarar variáveis/constantes/funções ex.: #declare x: Integer;

StrToInt() \\ converte um número passado como string para um número inteiro

Ex.:

#declare texto: String;

#declare numero: Integer;

texto = “155”;

numero = StrToInt(texto);

imprime(/numero)

StrToDouble \\ converte um número passado como String para um número Double

IntToStr \\ converte o valor de Inteiro para String

DoubleToStr \\ converte o valor de decimal para string

IntToDouble \\ converte o valor de inteiro para decimal

DoubleToInt \\ converte o valor de decimal para inteiro

Operadores Lógicos:

&& \\ utilizado para a operação lógica ‘E’

|| \\ utilizado para a operação logica ‘OU’

== \\ utilizado para verificação de igualdade

!= \\ utilizado para a verificação de não igualdade

> \\ utilizado para a verificação de MAIOR QUE

>= \\ utilizado para a verificação de MAIOR OU IGUAL A

< \\ utilizado para a verificação de MENOR QUE

<= \\ utilizado para a verificação de MENOR OU IGUAL A

Operador de comparação e condicional if:

Sintaxe:

if ({condição}) then

Begin

\\ aqui estará o código que será executado se a condição for favorável

End;

else

Begin

\\ aqui estará o código que será executado caso a condição não seja favorável

\\ note que deve conter um “ ; “ no final do bloco de execuções “ End; ”

End;

Ex.:

#declare x: Integer;

x = recebe(“Digite um número”); \\ o a string dentro do recebe() eh é impressa no terminal para o usuário

If (x == 10) then

Begin

Imprime(“X eh igual a 10”);

End;

Tipos de dados/ variáveis da MLP

Inteiro: Integer;

Decimal: Double;

Texto: String; \\ todas as strings devem ser escritas entre “”. Exemplo.: “meu texto”

Boleana: Bool; \\ números decimais devem ser escritos no formato Americano ex.:(10.5)

Assim como em C/C++ a deve-se conter um ponto e vírgula ( ; ) no final de cada expressão passada

Segue exemplo:

#declare x: Integer

#declare verifica: Bool;

\\ exemplo de como criar uma função caso queira utilizar

function NomeDaFuncao(parametros);

Begin

\\ código a ser executado da função ,

\\ note que o código a ser executado deve estar entre ‘Begin’ e ‘End;’ assim como no if

return valor; \\ utilizar o return caso deseje receber a resposta da função

End;

\\ para chamar a função basta escrever o nome da função e seus parâmetros

NomeDaFuncao(parâmetros);

\\ laço de repetição while

while (condição) then

Begin

\\ código a ser executando enquanto a condição for favorável

\\ note que o código a ser executado deve estar entre ‘Begin’ e ‘End;’ assim como no if

End;

Codigo realizando o exercício proposto em sala (idem ao código em python)

#declare frequenciaAluno, horasAula: Double; \\ declarando as variáveis a serem utilizadas

\\ função para calcular a frequencia

function calculaFrequencia(x,y):

Begin

return ( mathDiv(x\*100),y ); [\\ calculando](file:///\\calculando) a frequência do aluno

End;

\\ insert de valores

frequenciaAluno = recebe(“Digite o total de horas do aluno em sala”);

horasAula = recebe(“Digite o total de horas da aula”);

// comparação de resultado

If ( calculaFrequencia(frequenciaAluno, horasAula) < 75.0 ) then

Begin

Imprime(“Aluno reprovado por frequência em sala”);

End;

Else

Begin

Imprime(“Aluno aprovado pela frequência em sala”);

End;