

Repetição

1. Var

polegadas:inteiro

Início

para polegadas de 1 ate 20 faca

 escreval (polegadas, " polegada(s) é igual a: ", polegadas*2.54, " cm")

fimpara

escreval ("Fim")

Fimalgoritmo

2. Var

primeirotermo,razao,numerodetermos,contador:inteiro

Início

escreval ("Digite o primeiro termo da PA")

leia (primeirotermo)

escreval ("Digite a razão da PA")

leia (razao)

escreval ("Digite o termo que procura")

leia (numerodetermos)

escreval ("O 1 termo dessa PA vale:",primeirotermo)

para contador de 2 ate numerodetermos faca

 primeirotermo <- primeirotermo+razao

 escreval ("O", contador, " termo dessa PA vale:", primeirotermo)

fimpara

Fimalgoritmo

3. Var

limitesuperior,limiteinferior,contador,soma:inteiro

Início

escreval ("Digite o limite inferior")

leia (limiteinferior)

escreval ("Digite o limite superior")

leia (limitesuperior)

para contador de limiteinferior ate limitesuperior faca

 se contador % 2=0 entao

 soma <- contador+soma

 fimse

fimpara

escreval ("A soma total dos números pares é:", soma)

Fimalgoritmo

4. Var

numeroanterior,numeroatual,termos,novonumero,contador:inteiro

Início

escreval ("Digite o primeiro valor da Ricci")

leia(numeroanterior)

```

escreval("Digite o segundo valor da Ricci")
leia(numeroatual)
escreval("Insira o número de termos da Ricci")
leia(termos)
se termos <= 1 entao
    escreval ("Esses termos são inválidos, são necessários dois ou mais termos")
senao
    escreval ("O 1. número Ricci é:", numeroanterior)
    escreval ("O 2. número Ricci é:", numeroatual)
fimse
para contador de 3 ate termos faca
    novonumero <- numeroatual + numeroanterior
    numeroanterior <- numeroatual
    numeroatual <- novonumero
    escreval ("O", contador, ". Numero Ricci é:", novonumero)
fimpara

```

Fimalgoritmo

5. Var

numeroanterior,numeroatual,termos,novonumero,indice:inteiro

Inicio

```

escreval ("Digite o primeiro valor da Fetuccine")
leia(numeroanterior)
escreval("Digite o segundo valor da Fetuccine")
leia(numeroatual)
escreval ("Insira o número de termos da série")
leia(termos)
se termos <= 1 entao
    escreval ("Esses termos são inválidos, são necessários dois ou mais termos")
senao
    escreval ("O 1. número Fetuccine é:", numeroanterior)
    escreval ("O 2. número Fetuccine é:", numeroatual)
fimse
para indice de 3 ate termos faca
    se indice % 2=0 entao
        novonumero <- numeroatual - numeroanterior
    senao
        novonumero <- numeroatual + numeroanterior
    fimse
    numeroanterior <- numeroatual
    numeroatual <- novonumero
    escreval ("O", indice, "º Numero Fetuccine é:", novonumero)
fimpara

```

Fimalgoritmo

6. Var

limiteinferior,limitesuperior,contador1,n,divisores,contador2,primo:inteiro

Inicio

```
escreva("Insira o limite inferior: ")
leia(limiteinferior)
escreva("Insira o limite superior: ")
leia(limitesuperior)
para contador1 de limiteinferior ate limitesuperior faca
    n<-contador1
    divisores<-0
    para contador2 de 1 ate n faca
        se (n % contador2 = 0) entao
            divisores<-divisores+1
        fimse
    fimpara
    se (divisores = 2) entao
        escreval(contador1)
        primo<-primo+1
    fimse
Fimpara
```

Fimalgoritmo

7. Var

numeroentrada,contador:inteiro

Inicio

```
escreval ("Determine que será multiplicado")
leia (numeroentrada)
para contador de 1 ate 10 faca
    escreval (numeroentrada, " vezes", contador , " é igual a:", numeroentrada*contador)
fimpara
```

Fimalgoritmo

8. Var

numero,contador,contador1,contador2,divisores:inteiro

Inicio

```
numero<-1
enquanto n<>0 faca
    escreva("Digite um número: ")
    leia(numero)
    divisores<-0
    para contador2 de 1 ate n faca
        se (n mod contador2 = 0) entao
            divisores<-divisores+1
        fimse
    fimpara
    se (divisores = 2) entao
```

```
    contador<-contador+1
fimse
fimenquanto
escreval("Total de números primos lidos:",contador)
Fimalgoritmo
```

9. Var

numero,contador,index,potencia:inteiro

Inicio

```
numero<-1
enquanto numero<>0 faca
  escreva("Insira um número: ")
  leia(numero)
  para index de 1 ate numero faca
    potencia <- index*index
    se potencia=numero entao
      contador <- contador+1
  fimse
fimpara
fimenquanto
escreval("O número de quadrados perfeitos é: ",contador)
```

Fimalgoritmo

10. Var

dividendo,divisor,quociente:inteiro

Inicio

```
escreval ("Insira, respectivamente, o dividendo e o divisor")
leia (dividendo, divisor)
quociente <- 0
enquanto dividendo >= divisor faca
  dividendo <- dividendo - divisor
  quociente <- quociente+1
fimenquanto
escreval ("O quociente é: ", quociente)
```

Fimalgoritmo

11. Var

raiz,contador,impar:real

Inicio

```
escreval ("Delimite a raiz a ser calculada")
leia (raiz)
contador <- 0
impar <- 1
enquanto (raiz > 0 ) faca
  raiz <- raiz-impar
  impar <- impar+2
```

```
    contador <- contador+1
    escreval("O valor da raiz é:", raiz, " enquanto o contador é:", contador)
fimenquanto
escreval ("A raiz quadrada é mais ou menos:", contador)
Fimalgoritmo
```