Alunos:

Rafael Giordano Matesco

Rafael Sant'Ana de Andrade Osses

Davi Baruel Lara

2ª lista 1ºbimestre

A)visualG:

Algoritmo "semnome"

Var

nums: real

nums2: real

diferenca: real

Inicio

escreval ("escolha um numero")

leia (nums)

escreval ("escolha outro numero")

leia (nums2)

diferenca<-nums - nums2

escreval ("diferenca = ", diferenca)

Fimalgoritmo

B)visualG:

Algoritmo "semnome"

Var

num, ngt : inteiro

Inicio

escreval ("escolha um numero")

leia (num)

se (num>=1) entao

escreval("este numero é positivo", num)

senao

ngt<-num\*-1)

escreval("numero de sua escolha transformado em positivo",ngt)

fimse

Fimalgoritmo

C)visualG:

Algoritmo "semnome"

Var

num1, num2, num3, num4, med : real

Inicio

escreval ("digite a primeira nota")

leia (num1)

escreval ("digite a segunda nota")

leia (num2)

escreval ("digite a terceira nota")

leia (num3)

escreval ("digite a quarta nota")

leia (num4)

med<- (num1 + num2 + num3 + num4)/4)

se (med<5) entao

escreval ("reprovado")

escreval ("media final igual a:", med)

senao

escreval ("aprovado")

escreval ("media final igual a:", med)

fimse

fimalgoritmo

D)visualG:

E)visualG:

algoritmo "Bhaskara"

var

A, B, C, DELTA, x1, x2: real

inicio

escreval("Digite o valor de A: ")

leia(A)

escreval("Digite o valor de B: ")

leia(B)

escreval("Digite o valor de C: ")

leia(C)

DELTA <- (B\*B)-4\*A\*C

se(DELTA < 0) entao

escreval("Não há raízes reais")

senao

se(DELTA = 0) entao

x1 <- -b/2\*a

escreval("Uma única raiz real: x =", x1)

senao

se(DELTA > 0) entao

x1 <- (-b + raizq(DELTA))/(2\*a)

x2 <- (-b - raizq(DELTA))/(2\*a)

escreval("Duas raízes reais e distintas: ")

escreval("x1 =",x1, " x2 =",x2)

fimse

fimse

fimse

fimalgoritmo

G)visualG:

algoritmo "divisao"

var

valor1,valor2,valor3,valor4:inteiro

inicio

escreva ("Digite o primeiro valor = ")

leia (valor1)

escreva ("Digite o segundo valor = ")

leia (valor2)

escreva ("Digite o terceiro valor = ")

leia (valor3)

escreva ("Digite o quarto valor = ")

leia (valor4)

se ((valor1 mod 2)=0) e ((valor1 mod 3)=0) entao

escreval ("Primeiro valor é divisivel por 2 e 3 =",valor1)

escreval ("Dividido por 2 =",valor1/2)

escreval ("Dividido por 3 =",valor1/3)

fimse

se ((valor2 mod 2)=0) e ((valor2 mod 3)=0) entao

escreval ("Segundo valor é divisivel por 2 e 3 =",valor2)

escreval ("Dividido por 2 =",valor2/2)

escreval ("Dividido por 3 =",valor2/3)

fimse

se ((valor3 mod 2)=0) e ((valor3 mod 3)=0) entao

escreval ("Terceiro valor é divisivel por 2 e 3 =",valor3)

escreval ("Dividido por 2 =",valor3/2)

escreval ("Dividido por 3 =",valor3/3)

fimse

se ((valor4 mod 2)=0) e ((valor4 mod 3)=0) entao

escreval ("Quarto valor é divisivel por 2 e 3 =",valor4)

escreval ("Dividido por 2 =",valor4/2)

escreval ("Dividido por 3 =",valor4/3)

fimse

fimalgoritmo

H)visualG:

Algoritmo "semnome

var

a,b,c,d,f:inteiro

Inicio

Escreval ("H) ")

escreval ("Me fale quatro numeros: ")

escreva ("A= ")

leia (a)

se ( (a mod 2=0) ou (a mod 3=1)) entao

escreval ("numero divisivel por 2")

senao

escreval ("numero divisivel por 3")

fimse

escreva ("B= ")

leia (b)

se ( (b mod 2=0) ou (b mod 3=1)) entao

escreval ("numero divisivel por 2")

senao

escreval ("numero divisivel por 3")

fimse

escreva ("C= ")

leia (c)

se ( (c mod 2=0) ou (a mod 3=1)) entao

escreval ("numero divisivel por 2")

senao

escreval ("numero divisivel por 3")

fimse

escreva ("F= ")

leia (f)

se ((f mod 2=0) ou (f mod 3=1)) entao

escreval ("numero divisivel por 2")

senao

escreval ("numero divisivel por 3")

fimse

Fimalgoritmo

I)visualG:

Algoritmo "ex i 2 lista 1 bim"

Var

n1, n2, n3, n4, n5:inteiro

maior:inteiro

menor:inteiro

Inicio

escreval("digite um numero inteiro:")

leia(n1)

escreval("digite outro numero inteiro:")

leia(n2)

escreval("digite outro numero inteiro:")

leia(n3)

escreval("digite outro numero inteiro:")

leia(n4)

escreval("digite outro numero inteiro:")

leia(n5)

se((n1 > n2) e (n1 > n3) e (n1 > n4) e (n1 > n5))entao

maior <- n1

senao

se((n2 > n1) e (n2 > n3) e (n2 > n4) e (n2 > n5))entao

maior <- n2

senao

se((n3 > n1) e (n3 > n2) e (n3 > n4) e (n3 > n5))entao

maior <- n3

senao

se((n4 > n1) e (n4 >= n2) e (n4 > n3) e (n4 > n5))entao

maior <- n4

senao

se((n5 > n1) e (n5 >= n2) e (n5 > n3) e (n5 > n4))entao

maior <- n5

fimse

fimse

fimse

fimse

fimse

se((n1 < n2) e (n1 < n3) e (n1 < n4) e (n1 < n5))entao

menor <- n1

senao

se((n2 < n1) e (n2 < n3) e (n2 < n4) e (n2 < n5))entao

menor <- n2

senao

se((n3 < n1) e (n3 < n2) e (n3 < n4) e (n3 < n5))entao

menor <- n3

senao

se((n4 < n1) e (n4 < n2) e (n4 < n3) e (n4 < n5))entao

menor <- n4

senao

se((n5 < n1) e (n5 < n2) e (n5 < n3) e (n5 < n4))entao

menor <- n5

fimse

fimse

fimse

fimse

fimse

escreval("o maior numero e:",maior)

escreval("o menor numero e:",menor)

Fimalgoritmo

J)visualG:

Algoritmo "ex j 2 lista 1 bim"

Var

n:inteiro

Inicio

escreval("Digite um numero: ")

leia(n)

se(n%2 = 0)entao

escreval("e par")

senao

escreval("e impar")

fimse

Fimalgoritmo

K)visualG:

Algoritmo "ex k 2 lista 1 bim"

Var

n : inteiro

Inicio

escreval("faixa permitida:entre 0 e 9")

escreval("digite um valor: ")

leia(n)

se((n<0) ou (n>9))entao

escreval("seu numero esta fora da faixa permitida")

senao

escreval("seu numero esta dentro da faixa permitida")

fimse

Fimalgoritmo

L)visualG:

Algoritmo "ex l 2 lista 1 bim"

Var

n:inteiro

Inicio

escreval("digite um numero: ")

leia(n)

se(nao(n>3))entao

escreval(n," e menor que 3")

senao

escreval(n," e maior que 3")

fimse

Fimalgoritmo

M)visualG:

Algoritmo "ex k 2 lista 1 bim"

Var

sexo:inteiro

nome:caractere

Inicio

escreval("digite seu nome(pressione enter)")

escreval("e digite o seu sexo(use 1 para feminino e 2 para masculino):")

leia(nome)

leia(sexo)

se(sexo = 1)entao

escreval("Ilma Srta.",nome)

senao

se(sexo = 2)entao

escreval("Ilmo sr.", nome)

fimse

fimse

Fimalgoritmo

1ª lista 2ºbimestre

A)visualG:

Algoritmo "semnome"

Var

cont:inteiro

quadr:inteiro

Inicio

escreval("numeros de 15 a 200 ao quadrado")

//WHILE

cont <- 15

enquanto (cont <= 200) faca

quadr <- cont \* cont

escreval(quadr)

cont <- cont + 1

fimenquanto

//FOR

para cont de 15 ate 200 passo 1 faca

quadr <- cont \* cont

escreval(quadr)

fimpara

//DO-WHILE

cont <- 15

repita

quadr <- cont \* cont

escreval(quadr)

cont <- cont + 1

ate(cont = 201)

Fimalgoritmo

B)visualG:

Algoritmo "semnome"

Var

num:inteiro

tabu:inteiro

res:inteiro

Inicio

escreval("digite um numero qualquer")

leia(num)

//FOR

escreval("aqui esta a tabuada do seu numero:")

para tabu de 1 ate 10 passo 1 faca

res <- num \* tabu

escreval(num," X ",tabu," = ",res)

fimpara

escreval("")

escreval("")

escreval("digite um numero qualquer")

leia(num)

//WHILE

tabu <- 1

escreval("aqui esta a tabuada do seu numero:")

enquanto (tabu <= 10)faca

res <- num \* tabu

escreval(num," X ",tabu," = ",res)

tabu <- tabu + 1;

fimenquanto

escreval("")

escreval("")

escreval("digite um numero qualquer")

leia(num)

//DO-WHILE

tabu <- 1

escreval("aqui esta a tabuada do seu numero:")

repita

res <- num \* tabu

escreval(num," X ",tabu," = ",res)

tabu <- tabu + 1

ate (tabu = 11)

Fimalgoritmo

C)visualG:

Algoritmo "semnome"

Var

cont:inteiro

resultado:inteiro

Inicio

//FOR

escreva("a soma de todos os numeros de 1 a 100 = ")

para cont de 1 ate 100 passo 1 faca

resultado <- cont + 1

fimpara

resultado <- resultado \* 50

escreval(resultado)

escreval()

//WHILE

escreva("a soma de todos os numeros de 1 a 100 = ")

cont <- 1

resultado <- 1

enquanto (cont<=100)faca

resultado <- cont + 1

cont <- cont + 1

fimenquanto

resultado <- resultado \* 50

escreval(resultado)

escreval()

//DO-WHILE

escreva("a soma de todos os numeros de 1 a 100 = ")

cont <- 1

resultado <- 1

repita

resultado <- cont + 1

cont <- cont + 1

ate(cont=101) fimrepita

resultado <- resultado \* 50

escreval(resultado)

Fimalgoritmo

D)visualG:

Algoritmo "semnome"

Var

i:inteiro

n1:inteiro

Inicio

//FOR

para i de 2 ate 500 passo 2 faca

se(i%2=0)entao

n1 <- n1+i

fimse

fimpara

escreval("a soma de todos os pares de 1 a 500 e =",n1)

//WHILE

i <- 2

n1 <- 0

enquanto (i<=500)faca

se(i%2=0)entao

n1 <- n1+i

fimse

i <- i+2

fimenquanto

escreval("a soma de todos os pares de 1 a 500 e =",n1)

//DO-WHILE

i <- 2

n1 <- 0

repita

se(i%2=0)entao

n1 <- n1+i

fimse

i <- i+2

ate(i=501) fimrepita

escreval("a soma de todos os pares de 1 a 500 e =",n1)

Fimalgoritmo

E)visualG:

Algoritmo "semnome"

Var

c: inteiro

Inicio

ESCREVAL("E)0 a 20, numeros impares, comando (para).")

para c de 1 ate 20 passo 1 faca

se(c % 2=1)entao

escreval(c," numero impar")

senao

fimse

fimpara

escreval("Comando (enquanto):")

c<-1

enquanto (c<=20)faca

se(c % 2=1)entao

escreval(c," numero impar")

senao

fimse

c<-c+1

fimenquanto

escreval("Comando (Repita):")

c<-1

repita

se(c % 2=1)entao

escreval(c," numero impar")

senao

fimse

c<-c+1

ate (c>20)

Fimalgoritmo

F)visualG:

Algoritmo "semnome"

Var

c: inteiro

Inicio

escreval("F)Numeros divisiveis por 4 ate 200, com o comando (para).")

para c de 1 ate 200 passo 1 faca

se(c % 4=0)entao

escreval(c," numero DIVISIVEL por 4")

senao

fimse

fimpara

escreval("Comando (ENQAUNTO):")

c<-1

enquanto (c<=200)faca

se(c % 4=0)entao

escreval(c," numero DIVISIVEL por 4")

senao

fimse

c<-c+1

fimenquanto

escreval("Comando (Repita):")

c<-1

repita

se(c % 4=0)entao

escreval(c," numero DIVISIVEL por 4")

senao

fimse

c<-c+1

ate(c>200)

Fimalgoritmo

G)visualG:

Algoritmo "potencias"

var

c: inteiro

resultado: real

Inicio

escreval("G)Potencia de 3 ate o 15, com o comando (PARA).")

para c de 0 ate 15 passo 1 faca

resultado<- 3^c

escreval("3 elevado a ",c," = ",resultado)

fimpara

escreval("Comando (enquanto):")

c<-0

enquanto (c<=15)faca

resultado<- 3^c

escreval("3 elevado a ",c," = ",resultado)

c<-c+1

fimenquanto

escreval("Comando (repita):")

c<-1

repita

resultado<- 3^c

escreval("3 elevado a ",c," = ",resultado)

c<-c+1

ate(c>15)

Fimalgoritmo

H)visualG:

Algoritmo "potencia"

Var

resultado:real

n:inteiro

base: inteiro

expoente: inteiro

c:inteiro

Inicio

escreval("H)Potencia e base, com o comando (PARA).")

escreval("Deseja fazer quantas vezes:")

leia(n)

para c de 1 ate n passo 1 faca

escreva("Digite a base da potencia: ")

leia(base)

escreva("Agora coloque um expoente: ")

leia(expoente)

resultado<- base^expoente

escreval(base,"%d elevado ao ",expoente," = ", resultado)

fimpara

escreval("Comando (enquanto):")

escreva("Deseja fazer quantas vezes: ")

leia(n)

c<-1

enquanto (c<=n)faca

escreva("Digite a base da potencia: ")

leia(base)

escreva("Agora coloque um expoente: ")

leia(expoente)

resultado <- base^expoente

escreval(base," elevado ao ",expoente," = ", resultado)

c<-c+1

fimenquanto

escreval("Comando (repita):")

escreva("Deseja fazer quantas vezes: ")

leia(n)

c<-1

repita

escreva("Digite a base da potencia: ")

leia(base)

escreva("Agora coloque um expoente: ")

leia(expoente)

resultado<- base^expoente

escreval(base," elevado ao ",expoente," = ", resultado)

c<-c+1

ate(c<=n)

Fimalgoritmo

I)visualG:

J)visualG:

K)visualG:

L)visualG:

M)visualG:

Algoritmo "semnome"

Var

n, cont, soma:inteiro

media:real

Inicio

escreval("escreva um numero inteiro:")

para cont de 1 ate 10 passo 1 faca

leia(n)

soma <- soma + n

fimpara

media <- soma / 10

escreval("a soma de todos os numeros e:",soma)

escreval("a media de todos os numeros e:",media)

escreval("")

escreval("escreva um numero inteiro:")

soma<-0

media<-0

cont<-1

n <- 0

repita

leia(n)

soma <-soma + n

cont <- cont + 1

ate (cont=10)

escreval("a soma de todos os numeros e:", soma)

media <- soma / 10

escreval("a media de todos os numeros e:",media)

escreval("")

escreval("escreva um numero inteiro:")

soma <- 0

media <- 0

cont <- 1

enquanto (cont<=10) faca

leia(n)

soma <- soma + n

cont <- cont + 1

fimenquanto

escreval("a soma de todos os numeros e:",soma)

media <- soma / 10

escreval("a media de todos os numeros e:",media)

Fimalgoritmo

N)visualG:

Algoritmo "semnome"

Var

n, cont, soma, val:inteiro

media:real

Inicio

soma <- 0

media <- 0

val <- 0

enquanto (n>=0)faca

escreval("digite um numero:")

leia(n)

soma <- soma + n

val <- val + 1

se (n<0)entao

val <- val - 1

fimse

fimenquanto

media <- soma / val

escreval("a soma de todos os numeros positivos e: ",soma)

escreval("a media de todos os numeros positivos e: ",media)

escreval("o total de todos valores positivos lidos e: ",val)

n <- 1 soma <- 1 media <- 0 val <- 0

repita

escreval()

escreval("digite um numero:")

leia(n)

soma <- soma + n

val <- val + 1

se (n<0)entao

val <- val - 1

fimse

ate (n < 0)

media <- soma / val

escreval("a soma de todos os numeros positivos e: ",soma)

escreval("a media de todos os numeros positivos e: ",media)

escreval("o total de todos valores positivos lidos e: ",val)

Fimalgoritmo

O)visualG:

Algoritmo "semnome"

Var

n,cont, fat:inteiro

Inicio

fat <- 1

para n de 1 ate 10 passo 2 faca

fat <- fat \* n

fimpara

escreval("fatorial dos numeros impares entre 1 e 10: ",fat)

fat <- 1

n <- 1

enquanto (n<=10)faca

fat <- fat \* n

n <- n + 2

fimenquanto

escreval("fatorial dos numeros impares entre 1 e 10: ",fat)

fat <- 1

n <- 1

repita

fat <- fat \* n

n <- n + 2

ate(n=10)fimrepita

escreval("fatorial dos numeros impares entre 1 e 10: ",fat)

escreval()

Fimalgoritmo

P)visualG:

Algoritmo "aritmética"

Var

c,soma: inteiro

media: real

Inicio

escreval("P) Valores pares de 50 a 70, comando (para)")

escreval

para c de 50 ate 70 passo 1 faca

se(c % 2=0)entao

escreval(c," Valor Par!")

soma<- soma+c

media<- soma/11

senao

fimse

fimpara

escreval

escreval("Resultado da soma= ",soma)

escreval("Media: ",media)

escreval("Comando (enquanto):")

escreval

c<-50

enquanto (c<=70)faca

se(c % 2=0)entao

escreval(c," Valor Par!")

soma<-soma+c

media<-soma/11

senao

fimse

c<-c+1

fimenquanto

ESCREVAL

escreval("Resultado da soma= ",soma)

escreval("Media: ",media)

escreval

escreval("Comando (repita):")

escreval

soma<-0

c<-50

repita

se(c % 2=0)entao

escreval(c," Valor Par!")

soma<- soma+c

media<- soma/11

senao

fimse

c<- c+1

ate c>70

escreval

escreval("Resultado da soma= ",soma)

escreval("Media: ",media)

Fimalgoritmo

Q)visualG:

Algoritmo "potencia"

Var

resultado:real

n:inteiro

base: inteiro

expoente: inteiro

resp: caractere

comod: inteiro

lar:inteiro

area: real

comp: inteiro

larcomp: inteiro

Inicio

escreval("Q)Largura X Comprimento.")

ESCREVAL("Deseja comecar com o exercicio: s/n")

leia(resp)

repita

se(resp = "s")entao

escreval("Qual comodo da casa deseja calcular?")

leia(comod)

escreval("Coloque as informaçoes:")

escreva("Largura: ")

leia(lar)

escreva("Comprimento: ")

leia(comp)

area <- larcomp

escreval("Area do comodo = ", area)

escreval("Voce quer continuar: s/n ")

leia(resp)

fimse

ate(resp <> "s")

escreval("Comando (enquanto):")

escreval("Deseja comecar com o exercicio: s/n")

leia(resp)

enquanto (resp = "s")faca

se(resp = "s")entao

escreval("Qual comodo da casa deseja calcular?")

leia(comod)

escreval("Coloque as informaçoes")

escreva("Largura: ")

leia(lar)

escreva("Comprimento: ")

leia(comp)

area <- larcomp

escreval("Area do comodo = ", area)

escreval("Voce quer continuar: s/n ")

leia(resp)

fimse

fimenquanto

escreval("Quantas vezes deseja fazer no comando:PARA?")

leia(n)

para c de 1 ate n passo 1 faca

escreval("Qual comodo da casa deseja calcular?")

leia(comod)

escreval("Coloque as informaçoes:")

escreva("Largura: ")

leia(lar)

escreva("Comprimento: ")

leia(comp)

area <- lar\*comp

escreval("Area do comodo = ", area)

fimpara

Fimalgoritmo

R)visualG:

S)visualG: