Problema 1

Generar los procedimientos y/o funciones que impriman los valor pares, impares y el promedio de un arreglo bidimensional. El (los) procedimiento(s) o método(s) deben ser invocados desde el método principal (quien es el único responsable de gestionar las entradas/salidas); además el método debe recibir como parámetro un arreglo bidimensional. Para este ejercicio utilize Pseint, utilizando metodos y funciones uno para llenar la matriz datos de una matriz y otro para separar los valores pares

1.)Inicio

2.) Definir variables

3.) limF = 3

4.) limC= 3

5.) pares = ""

6.) impares = ""

7.) Dimensionar matriz[limF,limC]

8.) llenarMatriz(matriz, limF, limC)

1.) Para( i=0 i< limF-1 :i= i+1)

2.) Para( j=0 j< limC-1 :j = j+1)

3.) Escribir (matriz[i,j]) , " "Sin Saltar

4.) FinPara

6.) Escribir ""

7.) FinPara

8.) Escribir "Pares: ", separarPares(matriz, limF, limC, pares)

9.) Escribir "Impares: ", separarImpares(matriz, limF, limC, impares)

10.) FinAlgoritmo

1.) Metodo llenarMatriz(matriz, limF, limC)

2.) Para( i=0 i< limF-1 :i= i+1)

3.) Para( j=0 j< limC-1 :j = j+1)

4.) matriz[i,j]=Azar(1,10)

5.) FinPara

6.) FinPara

7.) FinSubProceso

1.) Funcion pares=separarPares(matriz, limF, limC, pares)

2.) Para( i=0 i< limF-1 :i= i+1)

3.) Para( j=0 j< limC-1 :j = j+1)

4.) Si (matriz[i,j] MOD 2)==(0)

5.) pares = concatenar(pares , ConvertirATexto((matriz[i,j])))

6.) pares = concatenar(pares," ")

7.) FinSi

8.) FinPara

9.) FinPara

10.) FinFuncion

1.)Funcion impares=separarImpares(matriz, limF, limC, impares)

2.) Para( i=0 i< limF-1 :i= i+1)

3.) Para( j=0 j< limC-1 :j = j+1)

4.) Si (matriz[i,j] MOD 2)<>(0)

5.) impares = concatenar(impares , ConvertirATexto((matriz[i,j])))

6.) impares = concatenar(impares," ")

7.) FinSi

8.) FinPara

9.) FinPara

10.)FinFuncion

11.)Fin

Prueba de Escritorio

8 3 0

4 6 5

2 1 3

Pares: 8, 0,4,6,2

Impares: 3,5,1,3

Problema 2

Generar una solución que implemente 3 procedimientos. Que permitan calcular el área de un cuadrado, área de un triángulo y área de un rectángulo. Cada procedimiento/función debe recibir los datos necesarios y generar el valor correspondiente. Se debe invocar a los procedimientos desde un método principal; Si el usuario ingresa 1 se llama al procedimiento obtenerAreaCuadrado; 2 se llama al procedimiento obtenerAreaTriangulo; 3 se llama al procedimiento obtenerAreaCuadrado.

El área del cuadrado es igual a lado x lado x lado x lado

El área del triángulo es igual a (base x altura)/2

El área del rectángulo es igual a base x altura

Realizare tres funciones las cuales cada una se encaragara de hacer los calculos respectivos, como el Area del Triangulo, Areadel Cuadrardo, Area del Rectangulo

1.) Inicio

2.) Metodo Principal()

3.) Definicion de Variables

n1, n2 como Real

opcion como Entero

4.) Escribir "[1] Calcular el área de un triángulo"

5.) Escribir "[2] Calcular el área de un cuadrado"

6.) Escribir "[3] Calcular el área de un rectángulo"

7.) Leer opcion

9.) Si opcion = 1 entonces

10.) Escribir "Ingrese la base del triángulo: "

11.) Leer n1

12.) Escribir "Ingrese la altura del triángulo: "

13.) Leer n2

14.) Escribir "El área del triángulo = " + areaTriangulo(n1, n2)

15.) Fin Si

16.) Si opcion = 2 entonces

17.) Escribir "Ingrese el lado del cuadrado: "

18.) Leer n1

19.) Escribir "El área del cuadrado = " + areaCuadrado(n1)

20.) Fin Si

21.) Si opcion = 3 entonces

22.) Escribir "Ingrese la base del rectángulo: "

23.) Leer n1

24.) Escribir "Ingrese la altura del rectángulo: "

25.) Leer n2

26.) Escribir "El área del rectángulo =" + areaRectangulo(n1, n2)

27.) Fin Si

1.) Función areaTriangulo(n1, n2)

2.) Declarar respuesta como Real

3.) respuesta = (n1 \* n2) / 2

4.) Devolver respuesta

5.) Fin Función

1.) Función areaCuadrado(n1)

2.) Declarar respuesta como Real

3.) respuesta = n1 ^ 2

4.) Devolver respuesta

5.) Fin Función

1.) Función areaRectangulo(n1, n2)

2.) Declarar respuesta como Real

3.) respuesta = n1 \* n2

4.) Devolver respuesta

5.) Fin Función

6.) Fin

Prueba de Escritorio

Triangulo:

[1] Calcular el area de un triangulo

[2] Calcular el area de un cuadrado

[3] Calcular el area de un rectangulo

1

Ingrese la base del triangulo:

4

Ingrese la altura lado del triangulo:

4

El area del triangulo= 8.0

Cuadrado:

[1] Calcular el area de un triangulo

[2] Calcular el area de un cuadrado

[3] Calcular el area de un rectangulo

2

Ingrese un lado del cuadrado:

7

El area del cuadrado= 49.0

Rectangulo:

[1] Calcular el area de un triangulo

[2] Calcular el area de un cuadrado

[3] Calcular el area de un rectangulo

3

Ingrese la base del rectangulo:

6

Ingrese la altura del rectangulo:

5

El area del rectangulo= 30.0

Problema 3

Generar una función que tenga 4 parámetros de tipo decimal y devuelva el promedio

cualitativo de los parámetros. Si el promedio es: De 0 a 5 el promedio cualitativo

es Regular De 5.1 a 8 el promedio es Bueno De 8.1 a 9 el promedio es Muy Bueno

De 9.1 a 10 el promedio es Sobresaliente. A la función se la debe llamar desde un

método principal. Los parámetros necesarios para llamar a la función, deben ser

ingresados solicitados al usuario.

1.) Inicio

2.) Definicion de variables

lim (4) como entero

promedio[lim] como Cadena(A-Z, a-z, " ")

3.) Metodo Principal()

4.) Definicion de variables

notas[lim] como Real(0-10)

5.) Para (i = 0; i < lim; i = i+1)

6.) Escribir "Ingrese la nota del estudiante ", (i+1)

7.) Leer notas[i]

8.) FinPara

9.) promCualitativo(notas, promedio)

10.) presentarPromCualitativo(promedio)

11.) Fin Metodo Principal

1.) Funcion promCualitativo(notas[], promedio[]) : d

2.) Para (i = 0; i < lim; i = i + 1)

3.) Si (notas[i] >= 0) Y (notas[i] <= 5)

4.) promedio[i] = "Regular"

5.) Sino (notas[i] >= 5.1) Y (notas[i] <= 8)

6.) promedio[i] = "Bueno"

7.) Sino (notas[i] >= 8.1) Y (notas[i] <= 9)

8.) promedio[i] = "Muy bueno"

9.) Sino

10.) promedio[i] = "Sobresaliente"

11.) FinSino

12.) FinSino

13.) FinSino

14.) Finsi

15.) Finpara

16.) FinFuncion promCualitativo

1.) Funcion presentarPromCualitativo(Promedio[] Como Cadena) Como Cadena

2.) Para (i = 0; i < lim; i = i + 1)

3.) Escribir "El estudiante numero ", (i+1), " tiene una nota ", promedio[i]

4.) FinPara

5.) FinFuncion presentarPromCualitativo

6.) Fin

Prueba de Escritorio

Ingrese la nota del estudiante 1

4.2

Ingrese la nota del estudiante 2

7.3

Ingrese la nota del estudiante 3

10

Ingrese la nota del estudiante 4

8.9

El estudiante numero 1 tiene una nota Regular

El estudiante numero 2 tiene una nota Bueno

El estudiante numero 3 tiene una nota Sobresaliente

El estudiante numero 4 tiene una nota Muy bueno

Problema 4

Generar un procedimiento para calcular el valor de la planilla de luz y otro procedimiento para calcular el valor del predio de un bien inmueble. Cada procedimiento debe tener 2 parámetros (tipo cadena para nombre del cliente, cédula del cliente).

En el procedimiento de planilla de luz se debe pedir los siguiente datos valor del kilowatio y el número de kilowatios del mes. Y se genera en pantalla el siguiente reporte: Cliente Ana Contreras con cédula 1100112233 debe cancelar el valor de $10

En el procedimiento del predio se debe pedir el valor de inmueble y el para obtener el valor del predio se saca el 2% del valor del inmueble. Y se genera el siguiente reporte:

Cliente Ana Contreras con cédula 1100112233 tiene un bien inmueble valorado en $30000 y tiene que pagar de predio $ 600.

En el método principal; si el usuario ingresa 1 se llama al procedimiento calcularValorLuz; 2 se llama al procedimiento calcularPredio. Los datos que se necesita en cada procedimiento se los debe ingresar por teclado. Generare dos funciones uno para el valor de la planita y el otro del predio, predio = inmueble \* 0.2, planilla kv\* kvM

1.) Inicio

2.) Metodo Principal()

3.) Definicion de Variables

opcion como Entero(0-n)

kilovatio, kilovMes, resultado1 = 0, resultado2 = 0, inmueble : como Real(0-n)

nombre, cedula como Cadena(A-Z, a-z, " ")

4.) Escribir "Ingrese su nombre y apellido "

5.) Leer nombre

6.) Escribir "Ingrese su cedula"

7.) Leer cedula

8.) Escribir "[1] Generar planilla de luz"

9.) Escribir "[2] Generar Predio de un bien"

10.) Leer opcion

11.) Si(opcion = 1)

12.) Escribir "Ingrese el valor del kilovatio: "

13.) Leer kilovatio

14.) Escribir "Ingrese el numero de kilovatios gastados en el mes"

15.) Leer kilovMes

16.) Escribir "Cliente ", nombre, " con cedula ", cedula, " debe cancelar el valor de: ", calcularPlanillaLuz(kilovatio, kilovMes, resultado1)

17.) FinSi

18.) Si(opcion = 2)

19.) Escribir "Ingrese el valor del inmueble: "

20.) Leer inmueble

21.) Escribir "Cliente ", nombre, " con cedula ", cedula, " y tiene que pagar de predio: ", calcularpredio(inmueble, resultado2)

22.) FinSi

1.) Funcion calcularPlanillaLuz(kilovatio Como Real, kilovMes Como Real, resultado1 Como Real) Como Real

2.) resultado1 = kilovatio\*kilovMes

3.) Devolver resultado1

4.) FinFuncion calcularPlanillaLuz

1.) Funcion calcularpredio

2.) resultado2 = inmueble\*0.02

3.) Devolver resultado2

4.) FinFuncion calcularpredio

5.) Fin

Prueba de Escritorio

Predio de un bien:

Ingrese su nombre y apellido

Jose Cuenca

Ingrese su cedula

1104867146

[1] Generar planilla de luz

[2] Generar Predio de un bien

2

Ingrese el valor del inmueble:

53000

Cliente Blanca Alvarado con cedula 1104867146 y tiene que pagar de predio : 10600

Planilla de Luz:

Ingrese su nombre y apellido

Blanca Alvarado

Ingrese su cedula

1102823505

[1] Generar planilla de luz

[2] Generar Predio de un bien

1

Ingrese el valor del kilovatio:

0.65

Ingrese el numero de kilovatios gastados en el mes

500

Cliente Blanca Alvarado con cedula 1102823505 debe cancelar el valor de: 325

Problema 5

Generar las funciones/métodos que devuelvan las suma, resta y multiplicación de un arreglo bidimensional cuadrado; mismo que se lo recibe como parámetro. Realizare metodos y una function para generar la matriz aleatoria

1.) Inicio

2.) Metodo Principal()

3.) Definicion de Variables

4.) limF = 3, limC = 3 como Entero (n-0)

5.) Entero matriz1[][] = nuevo Entero [limF][limC]

6.) Entero matriz2[][] = nuevo Entero [limF][limC]

7.) Entero matriz3[][] = nuevo Entero [limF][limC]

8.) Entero matriz4[][] = nuevo Entero [limF][limC]

9.) Entero matriz5[][] = nuevo Entero [limF][limC]

10.) generarMatriz(matriz1)

11.) generarMatriz(matriz2)

12.) Escribir "Matriz 1"

13.) Escribir devolverMatriz(matriz1)

14.) Escribir "Matriz 2"

15.) Escribir devolverMatriz(matriz2)

16.) sumaMatrices(matriz1, matriz2, matriz3)

17.) Escribir "Suma de matrices"

18.) Escribir devolverMatriz(matriz3)

19.) restaMatrices(matriz1, matriz2, matriz4)

20.) Escribir "Resta de matrices"

21.) Escribir devolverMatriz(matriz4)

22.) multMatrices(matriz1, matriz2, matriz5)

23.) Escribir "Multiplicacion de matrices"

24.) Escribir devolverMatriz(matriz5)

25.) FinMetodo Principal

1.) Funcion devolverMatriz(matriz [][] como Entero) como Entero

2.) Para (i = 0; i < matriz(longitud); i = i + 1)

3.) Para (j= 0; matriz[0](longitud); i = i + 1)

4.) matriz[i][j] = Azar(9(-9)

5.) FinPara

6.) FinPara

7.) Devolver cadena

8.) FinFuncion devolverMatriz

1.) Metodo sumaMatrices (matriz1[][] como Entero, matriz2[][] como Entero, matriz3[][] como Entero) como Entero

2.) Para (i = 0; i < matriz1(longitud); i = i + 1)

3.) Para (j= 0; matriz[0](longitud); j = j + 1)

4.) matriz3[i][j] = matriz1[i][j] + matriz2[i][j]

5.) FinPara

6.) FinPara

7.) FinMetodo sumaMatrices

1.) Metodo restaMatrices (matriz1[][] como Entero, matriz2[][] como Entero, matriz4[][] como Entero) como Entero

2.) Para (i = 0; i < matriz1(longitud); i = i + 1)

3.) Para (j= 0; matriz[0](longitud); j = j + 1)

4.) matriz3[i][j] = matriz1[i][j] - matriz2[i][j]

5.) FinPara

6.) FinPara

7.) FinMetodo restaMatrices

1.) Metodo multMatrices (matriz1[][] como Entero, matriz2[][] como Entero, matriz5[][] como Entero) como Entero

2.) Para (int i = 0; i < matriz1(longitud); i = i + 1)

3.) Para (int j = 0; j < matriz2[0](longitud); j = j + 1)

4.) matriz5[i][j] += matriz1[i][j] \* matriz2[i][j]

5.) FinPara

6.) FinPara

7.) FinMetodo multMatrices

8.) Fin

Prueba de Escritorio

