Ejercicio 1

Definición del problema: Obtener el resultado de una expresión sabiendo que para A=2 y B=5.

Análisis

Datos de entrada

A , B : Entero

Datos de salida

Resultado: Entero

Proceso

resultado = 3 \* A – 4 \* B / A ^ 2

Diseño

|  |
| --- |
| Entidad que resuelve: Expresiones aritméticas |
| Variables:  A, B : Enteros // Almacenan 2 valores que darán el resultado  Resultado: Entero // Almacena el resultado de la expresión aritmética |
| Nombre del algoritmo: expresión\_aritmética   1. leer A 2. Leer B 3. resultado 🡨 3\*A-4\*B/A^2 // realiza los cálculos y asigna el resultado 4. mostrar resultado |

Ejercicio 2

Definición del problema: Obtener el resultado de una expresión dada

Análisis

Datos de entrada

expresión : Real

Datos de salida

Resultado: Real

Proceso

resultado = 4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2

Diseño

|  |
| --- |
| Entidad que resuelve: Expresiones aritméticas |
| Variables:  expresion : real // expresion almacena la expresion aritmética a resolver y  resultado: float // almacena el el resultado obtenido luego de realizar los calculos |
| Nombre del algoritmo: expresión\_aritmética2   1. leer expresion 2. resultado 🡨 4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2 // realiza los cálculos y asigna el resultado 3. mostrar resultado |

Ejercicio 4

Definición del problema: Obtener las siguientes expresiones aritméticas

Análisis

Datos de entrada

expresión: Real

Datos de salida

Resultado: Real

Proceso

resultado = 4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2

Diseño

|  |
| --- |
| Entidad que resuelve: Expresiones aritméticas |
| Variables:  expresión : real // expresión almacena la expresion aritmética a resolver y  resultado: flota // almacena el el resultado obtenido luego de realizar los calculos |
| Nombre del algoritmo: expresión\_aritmética2   1. leer expresión 2. resultado 🡨 4 / 2 \* 3 / 6 + 6 / 2 / 1 / 5 ^ 2 / 4 \* 2 // realiza los cálculos y asigna el resultado 3. mostrar resultado |

Ejercicio 5:

Definición del problema: Evaluar las expresiones dado A, B y C

Análisis:

Datos de entrada:

A, B y C: enteros

Datos de salida:

resultadoA, resultadoB, resultadoC : reales

proceso:

resultadoA:B = A – B ^ 2 / 4 \* C

restultadoB = (A \* B) / 3 ^ 2

resultadoC = (((B + C) / 2 \* A + 10) \* 3 \* B) – 6

Diseño:

|  |
| --- |
| Entidad que resuelve: Expresiones aritméticas |
| Variables  a, b, c : enteros  resultadoA, resultadoB, resultadoC : float |
| Nombre del algoritmo: calculo\_de\_expresiones  Inicio  Leer a  Leer b  Leer c  resultadoA 🡨 B \* A – B ^ 2 / 4 \* C  resultadoB 🡨(A \* B) / 3 ^ 2  resultadoC🡨(((B + C) / 2 \* A + 10) \* 3 \* B) – 6  mostrar resultadoA  mostrar resultadoB  mostrar resultadoC  fin |

Ejercicio 6

Análisis

Definición del problema: evaluar el resultado dado x = 3, y = 4, z= 1;

Datos de entrada:

x, y, z: enteros

datos de salida:

r1: entero

r2: boolean

Proceso:

R1 = y + z

R2 = x >= R1

|  |
| --- |
| Entidad que resuelve: expresión algebraica |
| Variables  x, y, z : enteros  R1: enteros  R2: boolean |
| Nombre del algoritmo: evaluar\_resultados  Inicio  Leer x  Leer y  Leer z  R1 🡨 y + z  R2 🡨 x>= R1  Mostrar R1  Mostrar R2  Fin |

Ejercicio 7

Análisis

Definición del problema: obtener los resultados sabiendo que contador1 = 3 y contador2 = 4

Datos de entrada:

Contador1, contador2: enteros

Datos de salida:

R1 entero

R2 boolean

Procesos:

R1 = ++ contador

R2= contador1 < contador2

|  |
| --- |
| Entidad que lo resuelve: expresiones Aritméticas |
| Variables  Contador1, contador2: enteros  R1: entero  R2: boolean |
| Nombre del algoritmo: obtener\_ resultados  Inicio  Leer contador1  Leer contador2  R1 🡨 ++contador1  R2🡨 contador1<contador2  Mostrar contador1  Mostrar contador2  Fin |

Ejercicio 8

Fase de Análisis

Definición del problema: Evaluar resultados sabiendo que a = 31, b= -1, x =3, y=2

Datos de entrada:

a, b, x, y: enteros

Datos de salida:

resultado: real

Proceso:

resultado = a+b-1 < x \*y

Fase de diseño:

|  |
| --- |
| Entidad que lo resuelve expresiones algebraicas / lógicas |
| Variables  a, b, x, y: entero  resultado: boolean |
| Nombre del algoritmo: obtener\_resultado  Inicio  Leer a  Leer b  Leer x  Leer y  Resultado 🡨 **a+b-1 < x \*y**  *mostrar* resultado  Fin |

Ejercicio 9

Fase de análisis

Definición del problema: obtener el resultado sabiendo que x =6, y =8

Datos de entrada:

x, y: entero

Datos de salida

resultado: boolean

Proceso

Resultado = !(x<5) && !(y>=7)

|  |
| --- |
| Entidades que lo resuelven: expresiones lógicas |
| Variables  x, y: entero  Resultado boolean |
| Nombre del Algoritmo: obtener\_resultaod  Inicio  Leer x  Leer y  Resultado 🡨 !(x<5) && !(y>=7)  *Mostrar* resultado  Fin |

Ejercicio 10

Fase de análisis

Descripción del problema: obtener los resultados, sabiendo que i = 22 y j=3

Datos de entrada

i, j = enteros

Datos de salida

resultado: boolean

Proceso:

!((i>4) || !(j<=6))

Fase de diseño:

|  |
| --- |
| Entidad que lo resuelve: expresiones lógicas |
| Variables  i, j: enteros  resultado: boolean |
| Nombre del algoritmo: obtener\_resultados  Inicio  Leer I  Leer j  Resultado 🡨 !((i>4) || !(j<=6))  *Mostrar* resultado  Fin |

Ejercicio 11

Fase de análisis

Descripción del problema: obtener los resultados sabiendo que a = 34b =12 y c= 8

Datos de entrada:

a, b, c: enteros

Datos de salida:

resultado: boolean

Proceso

Resultado = !(a+b==c) || (c!=0)CC(b-c>=19)

|  |
| --- |
| Entidad que lo resuelve: expresiones lógicas |
| Variables:  a, b, c: enteros  resultado: boolean |
| Descripción del algoritmo: obtener\_resultado  Inicio  Leer a  Leer b  Leer c  Resultado 🡨 !(a+b==c) || (c!=0)CC(b-c>=19)  Mostrar resultado  Fin |

Ejercicio 12

Fase de Análisis

Descripción del problema: saludar con el nombre brindado

Datos de entrada:

Nombre: string

Datos de salida

Saludo: string

Procesos

Saludar con el nombre ingresado

Fase de diseño

|  |
| --- |
| Entidad que lo resuelve: consola |
| Variables  Nombre string  Saludo string |
| Nombre del algoritmo: Nombre\_saludo  Inicio  Nombre 🡨ingresar  Leer nombre  *Mostrar* saludo + nombre  Fin |

Ejercicio 13

Descripción del problema: Calcular el perímetro y área de un rectángulo sabiendo su base y altura

Datos de entrada:

base, altura: entero

datos de salida

área, perímetro: entero

procesos:

área = base \* altura

perímetro = base \*2 + altura \* 2

|  |
| --- |
| Entidad que lo resuelve: Expresiones aritméticas |
| Variables:  base, altura: entero  area, perímetro: entero |
| Nombre del algoritmo: calculos\_rectangulo  Inicio  leer altura  leer base  area 🡨base \* altura  perímetro 🡨base \*2 + altura \* 2  *mostrar* area  *mostrar* perímetro |

Ejercicio 14

Fase de análisis

Definición del problema: Obtener la hipotenusa de un triangulo rectángulo conociendo sus catetos

Datos de entrada:

catetoA, catetoB: entero

datos de salida:

hipotenusa: real

procesos

hipotenusa = √(catetoA) ^2 + (catetoB) ^2

Fase de diseño

|  |
| --- |
| Entidad que lo resuelve: Teorema de Pitágoras |
| Variables  catetoA, catetoB: entero  hipotenusa: real |
| Nombre del algoritmo: calculo\_hipotenusa  Inicio  Leer catetoA  Leer catetoB  hipotenusa 🡨pow (pow(catetoA,2) + pow (catetoB,2), 0.5)  mostrar hipotenusa  fin |

Ejercicio 15

Fase de análisis

Definicion del problema: Conociendo 2 numeros obtener suma, resta, multiplicación y división de estos.

Datos de entrada:

num1, num2: enteros

Datos de salida:

suma, resta, multiplicación, división: entero

procesos

suma = num1 + num2

resta = num1 – num2

multiplicación= num1\*num2

división = num1/num2

|  |
| --- |
| Entidad que lo resuelve: expresiones aritméticas |
| Variables  num1, num2: enteros  suma, resta, multiplicación, división: enteros |
| Definición del algoritmo: operaciones\_mate  Inicio  Leer num1  Leer num2  Suma 🡨num1 + num2  Resta 🡨 num1 – num2  Multiplicación 🡨 num1\* num2  Si num2 != 0 hacer  division = num1 / num2  sino  *mostrar*  “error”  fin si  *mostrar* suma  *mostrar* resta  *mostrar* multplicacion  *mostrar* division  fin |

Ejercicio 16

Definición del problema: Convertir de temperatura Farenheit a ˚˚c

Datos de entrada

temperaturaFarenheit: real

datos de salida

temperaturaC: real

proceso:

temperaturaC = (temperaturaFarenheit - 32) /1.8

fase de diseño

|  |
| --- |
| Entidad que lo resuelve: entidades aritméticas |
| Variables  temperaturaC, temperaturaFarenheit: real |
| Nombre del algoritmo: convertir\_temp  Leer temperaturaFarenheit  temperaturaC 🡨 (temperaturaFarenheit - 32) /1.8  *mostrar* TemperaturaC |

Ejercicio 17

Definicion del problema: Calcular la distancia entre el personaje y el power up

Análisis

Datos de entrada

coordenadasPj, coordenadasPowerUp : cordenadas cartesianas

radio, ancho, alto : enteros

datos de salida

Rectangulo y circulo dibujados en el lienzo

Procesos:

Calcular la distanci entre el rectángulo y circulo

Diseño

Ejercicio 21

Análisis:

Descripción del problema: Dibujar rectángulo usando estructuras iterativas

Análisis

Datos de entrada:

coordenadaRect: coordenadas cartesianas

ancho, alto, distanciaEntreRect: entero

Datos de salida

Los rectángulos dibujados en el lienzo

Proceso:

Dibujar los rectángulos

Diseño

|  |
| --- |
| Entidad: Lienzo |
| Variables:  coordenadasRect: coordenadas  ancho, alto, distanciaEntreRect: enteros |
| Nombre del algoritmo: dibujar\_rectangulos  Algoritmo:  Inicio}  anchoLienzo🡨 440  altoLienzo🡨420  distanciaEntreRect 🡨20  ancho 🡨40  alto🡨20  para x 🡨 coordenadasRect.x hasta anchoLienzo con paso (ancho+ distanciaEntreRect)  para y 🡨coordeandasRect.y hasta altoLienzo con paso (alto+ distanciaEntreRect)  dibujar un rectángulo en (x, y, ancho, alto)  fin para  fin para  fin |