Texto

Descripción generada automáticamente

RTA:

El resultado1 es igual a -21 porque primero se empieza por las operaciones de mayor precedencia que sería la multiplicación, ósea 5 x (-4) = -20. Luego seguiría hacer el resultado anterior sumado a la var1 y quedaría: 2+(-20) = -18. Y finalmente hacemos la resta entre lo que obtuvimos con 3, quedando así: **-21**

El resultado2 es igual a -21, acá se enfatiza o se provoca a que se resuelva lo que está entre paréntesis, cosa que es redundante porque el compilador es a lo primero que va a atacar. El procedimiento es el mismo que el anterior

El resultado3 es igual a -31, primero se realiza lo que esta entre paréntesis dando como resultado 7, luego a eso por orden de precedencia lo multiplicamos con var3, y quedaría 7\*(-4), ósea -28. Y finalmente a lo que obtuvimos hay que restarle 3. Y queda -31

El resultado4 es igual a -33, primero se hace lo de paréntesis dando como resultado -7, a eso ultimo lo multiplicamos por la var2 dando así, -35. Y al final a -35 le sumamos 2, quedando -33.

Lo que se imprime por pantalla es el primer if, porque la condición planteada da verdadero y la explicación ya fue dada.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Voy a hacer la tabla de la verdad para hacer más visual la explicación

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| X | Y | ¡X (lo opuesto) | ¡X||Y (en la disyunción obtenemos falso si ambas proposiciones son falsas) | (¡X||Y)&&X (en la conjunción si ambas prop. Son verdaderas, el resultado es V) |  |
| V | V | F | V | V |  |
| V | F | F | F | F |  |
| F | V | V | V | F |  |
| F | F | V | V | F |  |

Ahora analicemos item por item:

* La a) es falsa porque tanto X como Y deben tener el mismo valor para que se cumpla o de verdadera la condición lógica final
* La b) es verdadera, ya que se cumple y por demostración de la tabla de verdad es correcto.
* La c) es verdadera, pero un poco ambigua porque el resultado final no va a depender 100% de que sólo una de las 2 proposiciones sea verdadera.
* La d) es incorrecta, si x es falso, obtendremos un F

Texto

Descripción generada automáticamente

using System;

using System.Collections.Generic;

public class Hello{

static void mostrarVector(int[] vector)

{

for(int i = 0; i< vector.Length; i++)

{

Console.WriteLine(vector[i] + " ");

}

}

static bool esDivisiblePor5(int numero)

{

if(numero % 5 == 0)

{

return true;

}

return false;

}

static int sumandoLosDivisiblesPor5(int[] vectorDeNumeros)

{

int sumadorDeDivPor5 = 0;

for(int i = 0; i<vectorDeNumeros.Length; i++)

{

if(esDivisiblePor5(vectorDeNumeros[i]))

{

sumadorDeDivPor5 += vectorDeNumeros[i];

}

}

return sumadorDeDivPor5;

}

public static void Main(){

// Your code here!

int[] testingNumeros = {10, 2, -3, 5, 6, -5, 2, -5, 7, 5, 1, 5};

int respuesta = 0;

respuesta = sumandoLosDivisiblesPor5(testingNumeros);

Console.WriteLine(respuesta);

}

}

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

using System;

using System.Collections.Generic;

public class Hello{

static bool esSubconjunto(int[] A, int[] B)

{

int cantidadDeAciertos = 0;

for(int i=0; i<A.Length; i++) //El subconjunto generalmente es el más chico o igual al conjunto más grande

{

for(int j=0; j<B.Length; j++)

{

if(A[i] == B[j])

{

cantidadDeAciertos++;

}

}

}

//Ahora si la cantidad de aciertos coincide con la cantidad total de elementos del subconjunto es porque lo es

if(cantidadDeAciertos == A.Length)

{

return true;

}

return false;

}

public static void Main(){

// Your code here!

int[] conjuntoPadre = {10, 2, 3, 4, 100, 5, 1000};

int[] subconjunto = {10, 100, 1000};

bool respuesta = false;

respuesta = esSubconjunto(subconjunto, conjuntoPadre);

Console.WriteLine(respuesta);

}

}

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

using System;

using System.Collections.Generic;

public class Hello{

static void mostrarLista(List<int> A)

{

Console.Write("<");

for (int i = 0; i < A.Count; i++)

{

Console.Write(A[i]);

if (i < A.Count - 1)

Console.Write(", ");

else

Console.WriteLine(">");

}

}

static List<int> ejercicio5(List<int> lista)

{

List<int> resultado = new List<int>();

for(int i = 0; i< lista.Count; i++)

{

if(i==0)

{

if(lista[i]>lista[i+1])

{

resultado.Add(lista[i]);

}

}

if(i== (lista.Count-1))

{

if(lista[i]<lista[i-1])

{

resultado.Add(lista[i]);

}

}

if(i>0 && i<lista.Count-1)

{

if((lista[i]<lista[i-1]) && (lista[i]>lista[i+1]))

{

resultado.Add(lista[i]);

}

}

}

return resultado;

}

public static void Main(){

// Your code here!

List<int> A = new List<int> {7,5,3,1};

mostrarLista(ejercicio5(A));

//mostrarLista(A);

}

}

Texto

Descripción generada automáticamente

Acá van los métodos:

static bool esMayorA7YMenorA15(int num)

{

if(num > 7 && num < 15)

{

return true;

}

return false;

}

static void eliminadoLosMayoresA7YMenoresA15ConRemoveAll(List<int> A)

{

A.RemoveAll(esMayorA7YMenorA15);

}

static void eliminadoLosMayoresA7YMenoresA15SinRemoveAll(List<int> A)

{

for(int i = 0; i< A.Count; i++)

{

if(esMayorA7YMenorA15(A[i]))

{

A.Remove(A[i])

}

}

}