

Soluciones en versión PseInt.

1. Realiza un algoritmo que lea por teclado dos números enteros positivos y compruebe si dichos números son primos entre sí.

```
Algoritmo primosEntreSi
    Definir num1, num2 Como Entero
    Definir numPeq Como Entero
    Definir i como Entero
    Definir encontrado Como Logico
    Definir divisores Como Caracter

    Definir versionAEjecutar Como Entero

    versionAEjecutar = 2

    Repetir
        Escribir "Introduce el primer número positivo mayor que 0"
        Leer num1
    Hasta Que num1 >= 0

    Repetir
        Escribir "Introduce el segundo número positivo mayor que 0"
        Leer num2
    Hasta Que num2 >= 0

    // Ya tenemos aquí los dos números leídos.
    // Ahora, lo que haremos será buscar sus divisores y comprobar si
    alguno coincide

    // Existen 2 versiones, una que encuentra sólo el primer divisor
    común y se para ahí, más sencilla.
    // La segunda versión comprueba todos los divisores

    // Para hacerlo más eficiente solo recorreremos hasta el número más
    pequeño (a partir de ahí ya no habrá más divisores del número más
    pequeño y por tanto no tiene sentido)
    // Necesitamos comprobar también el propio número pequeño, pues por
    ejemplo 7 y 21 no son primos entre sí, pues ambos son divisibles por 7
    // Podríamos mejorarlo, pues desde la mitad del número pequeño hasta
    el propio número pequeño ya no habrá más divisores (7 ya no será
    divisible por 4 ni por un número mayor hasta llegar al propio 7). Esto
    se deja como desafío personal

    numPeq = num1
```

```

si num1 > num2 Entonces
    numPeq = num2
FinSi

si versionAEjecutar == 1 Entonces

    //Versión 1:
    encontrado = Falso
    i = 2
    mientras !encontrado y i <= numPeq
        //Si ambos son divisibles (resto = 0) por el mismo número,
        ya descartamos la posibilidad de que sean primos entre sí
        si num1%i == 0 y num2%i == 0 Entonces
            encontrado = Verdadero
        FinSi
        i = i + 1
    FinMientras

    si encontrado Entonces
        Escribir "Los números " num1 " y " num2 " no son primos
entre sí ya que ambos son divisibles por " (i - 1)
    SiNo
        Escribir "Los números " num1 " y " num2 " son primos
entre sí"
    FinSi
FinSi

si versionAEjecutar == 2 Entonces
    //Versión 2:
    // Nota que aquí no utilizamos la variable encontrado para
detener el bucle, pues queremos encontrarlos todos y no parar en el
primero
    encontrado = Falso // La utilizaremos para poder escribir el
mensaje correcto al finalizar la ejecución
    Para i=2 Hasta numPeq Con Paso 1 Hacer
        // La comprobación es la misma, pero lo que haremos será
añadir el texto a una variable que contendrá todos los divisores
anteriores
        si num1%i == 0 y num2%i == 0 Entonces
            // Todavía puede mejorarse para no dejar esa coma
final, pero esto lo dejaremos para cuando trabajemos con cadenas
            divisores = divisores + " " + ConvertirATexto(i) +
", "
        encontrado = Verdadero
    FinSi
Fin Para

```

```

        si encontrado Entonces
            Escribir "Los números " num1 " y " num2 " no son primos
entre sí ya que ambos son divisibles por " divisores
        SiNo
            Escribir "Los números " num1 " y " num2 " son primos
entre sí"
        FinSi
    FinSi
FinAlgoritmo

```

2. Realiza un algoritmo que calcule la sucesión de Fibonacci hasta el nivel X, siendo X un número entero positivo que se pedirá por teclado.

```

Algoritmo fibonacci
    Definir num, ant1, ant2 Como Entero
    Definir i como Entero
    Definir sucesion Como Caracter

    // El número debe ser positivo
    Repetir
        Escribir "¿Hasta qué nivel debo mostrar la sucesión de
fibonacci?"
        Leer max
        Hasta Que max > 0

        //Existen varias versiones. Una de las más famosas utilizando
recursividad. Pero aquí sólo veremos la versión que usa bucles
        ant2 = 0
        ant1 = 1

        // Los 2 primeros casos son casos base, por lo que no los
calcularemos
        si max == 1 Entonces
            sucesion = "0"
        SiNo
            sucesion = "0, 1, "
            si max > 2 Entonces
                // El primer número calculado de Fibonacci es el que ocupa
la tercera posición
                Para i = 3 Hasta max Con Paso 1 Hacer
                    // Y se calcula realizando la suma de los dos números

```

```
anteriores
    fib = ant2 + ant1
    // Por lo que ahora, tocará actualizar los números
anteriores
    ant2 = ant1 // Ahora ant2 vale lo que antes valía
ant1
    ant1 = fib // Y ant1 toma el valor del número recién
calculado

    sucesion = sucesion + ConvertirATexto(fib) + ", " //
Ya solo queda concatenerlo al resultado
    Fin Para
    FinSi
FinSi

    Escribir "La sucesión pedida es " + sucesion

FinAlgoritmo
```