

Presentación de la 4ª Práctica de PL para el Miércoles 13 de Enero 2021.

Para los que tengan como estructura complejas las pilas, o las listas, o los conjuntos:

Confeccionar el siguiente algoritmo expresado en vuestro lenguaje:

(a) (2,75) Completo (anidamiento de funciones/procedimientos, estructuras de datos complejas).

Pida por teclado un número N , y debe llamar a una función/procedimiento, denominada *fibonacci* con argumento N que deberá obtener los N primeros elementos de la serie de Fibonacci e introducirlos en una pila, o listas o conjunto denominada *Dato*. A su vez, anidada en *fibonacci* debe definirse otra función/procedimiento sin argumentos denominada *fibonacciimprime* que imprima los elementos de *Dato1* creado en *fibonacci*, así como, imprimir *Dato2*, donde $Dato2 = Dato * 2$, e imprimir *Dato3* = a la concatenación/unión de *Dato1* y *Dato2*. En el caso de pilas, crear *Dato3* como *Dato1* junto con la inserción uno a uno de los elementos de *Dato2*.

(b) (1,75) Anidamiento y datos elementales.

Lo mismo, pero sin crear la estructura *Dato1*, *Dato2* y *Dato3*. La función/procedimiento *fibonacci* debe definir a *fibonacciimprime* sin argumentos. *fibonacci* debe obtener los N primeros elementos de la serie de Fibonacci en una variable denominada *Dato* y llamar a *fibonacciimprime* elemento a elemento para imprimirlos. Además al final *fibonacci* debe imprimir el último elementos de la serie multiplicado por 10 y dividido por el $(N-1)$.

(c) (2,25) Solo estructuras de datos.

Lo mismo que el completo (a) sin definir funciones/procedimientos pero la generación de la serie de Fibonacci y las impresiones y operaciones se hacen en el programa principal.

(d) (1,25) Solo estructuras de datos elementales.

Lo mismo que (b) pero sin definición de funciones como (c).

Para los que tengan arrays

Confeccionar el siguiente algoritmo expresado en vuestro lenguaje:

(a) (2,75) Completo (anidamiento de funciones/procedimientos, estructuras de datos complejas).

Pida por teclado un número N , y debe llamar a una función/procedimiento, denominada *fibonacci* con argumento N que deberá obtener los 10 primeros elementos de la serie de Fibonacci e introducirlos en una array denominada *Dato[10]*. A su vez, anidada en *fibonacci* debe definirse otra función/procedimiento sin argumentos denominada *fibonacciimprime* que imprima los elementos de *Dato1* creado en *fibonacci*, así como, imprimir *Dato2*, donde $Dato2 = Dato * N$. En *fibonacci* al final debe crear la array *Dato3[N,2]* y *Dato4[2,N]*, donde la 1ª columna de *Dato3* debe tener los elementos de *Dato1* y la 2ª columna debe tener los elementos de *Dato2*. Y *Dato4* debe ser la traspuesta de *Dato3*. Al final deberá imprimir *Dato3*Dato4* y *Dato4*Dato3*.

(b) (1,75) Anidamiento y datos elementales.

La función/procedimiento *fibonacci* debe leer un número N y definir a *fibonacciimprime* sin argumentos. *fibonacci* debe obtener los N primeros elementos de la serie de Fibonacci en una variable denominada *Dato* y llamar a *fibonacciimprime* elemento a elemento para imprimirlos. Además al final *fibonacci* debe imprimir el último elementos de la serie multiplicado por 10 y dividido por el $(N-1)$.

(c) (2,25) Solo estructuras de datos.

Lo mismo que el completo (a) pero sin funciones/procedimientos y la generación de la serie de Fibonacci y las impresiones y operaciones se hacen en el programa principal.

(d) (1,25) Solo estructuras de datos elementales.

Lo mismo que (b) pero sin definición de funciones como (c).