



aprenderaprogramar.com

Ejercicio ejemplo resuelto de uso de arrays (arreglos) dinámicos en pseudocódigo I (CU00214A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº13 del Curso Bases de la programación Nivel II

24

EJERCICIO

Desarrollar el pseudocódigo y el diagrama de flujo para un programa con las siguientes opciones:

- a) Introducir un valor entero impar comprendido entre 1 y 19
- b) Calcular la serie numérica $1 + 3 + 5 + \dots + n$
- c) Calcular $1 * 3 * 5 * \dots * n$
- d) Salir del programa.

Nota: Los cálculos se han de realizar a través de un vector que almacene los valores impares comprendidos entre 1 y n .



SOLUCION:

PROGRAMA IMPARES01 [Pseudocódigo aprenderaprogramar.com]

Variables

Enteras: E, Dato, Cantidad, Valor()

1. Inicio

2. Mientras E <> 4 Hacer

2.1 Si E = 1 Entonces

 Llamar Genera

FinSi

2.2 Si E = 2 y Dato <> 0 Entonces

 Llamar CalculaSuma(Cantidad) PorValor

FinSi

2.3 Si E = 3 y Dato <> 0 Entonces

 Llamar CalculaMult(Cantidad) PorValor

FinSi

2.3 Si E = 3 y Dato <> 0 Entonces

Llamar CalculaMult(Cantidad) PorValor

FinSi

2.4 Mostrar "Elija opción: 1. Introducir dato

2. Calcular $1 + 3 + 5 + \dots + n$

3. Calcular $1 * 3 * 5 * \dots * n$

4. Salir"

2.5 Pedir E

Repetir

3. Fin

Módulo Genera

Variables

Enteras: Impar, i

Booleanas: Par

1. Dato = 0

2. Mientras Dato <= 0 ó Dato > 19 ó Par = Verdadero Hacer

Par = Falso

Mostrar "Introduzca un número impar entre 1 y 19"

Pedir Dato

Si Dato mod 2 = 0 Entonces

Par = Verdadero : Mostrar "El número debe ser impar"

FinSi

Repetir

3. Impar = - 1 : i = 0

4. Hacer

i = i + 1

Impar = Impar + 2

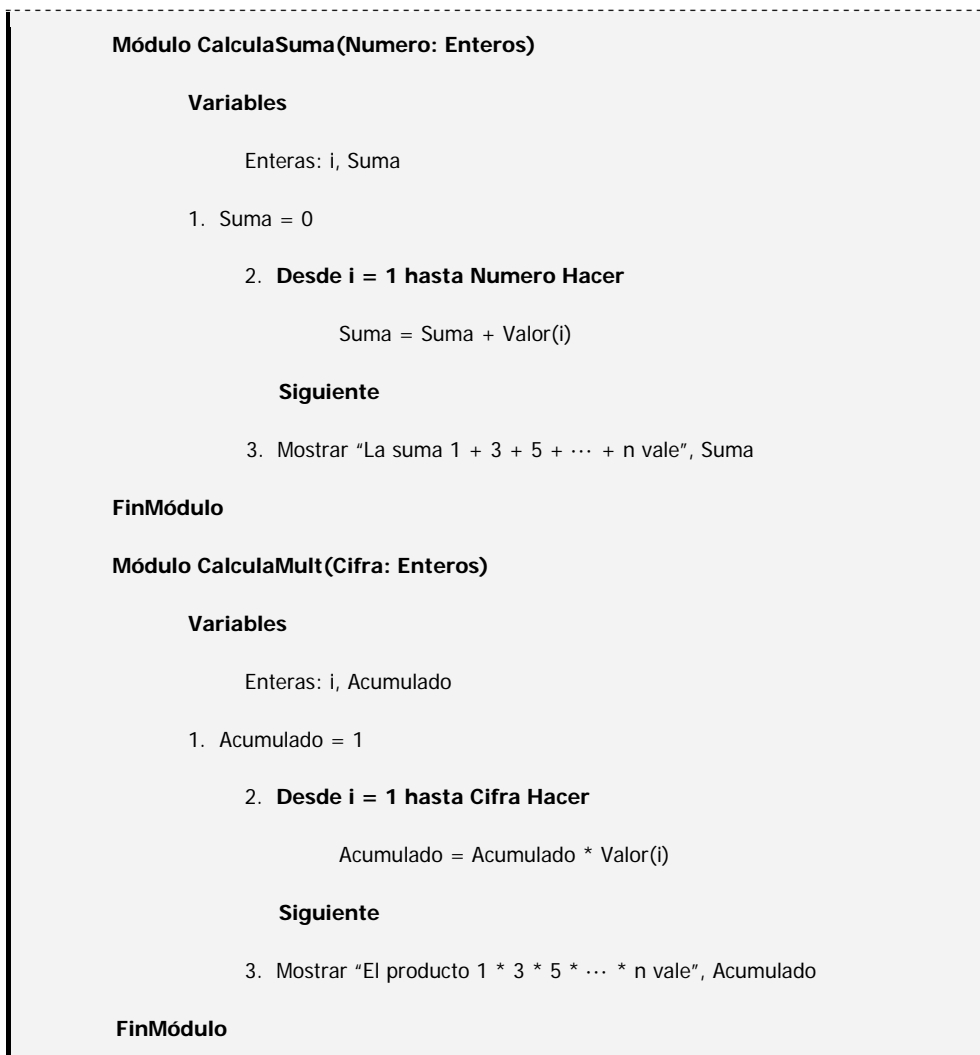
Redimensionar Valor(i)

Valor(i) = Impar

Repetir Mientras Impar < Dato

5. Cantidad = i

FinMódulo



Comentarios: El programa consta del algoritmo principal y 3 módulos. En el módulo Genera se pide un número impar entre 1 y 19 no permitiendo la salida hasta que se tiene un dato válido. Tras ello se generan una serie de valores: Valor(1), Valor(2), ... que contienen los números impares comprendidos entre el 1 y el proporcionado por el usuario.

Obsérvese que por cada impar que se procesa en el bucle se produce un redimensionamiento de la matriz Valor. El proceso es algo así:

Detectar valor → Redimensionar matriz → Asignar valor → Repetir si procede

Esto no es demasiado eficiente ya que nos obliga a redimensionar en cada paso del bucle. Una alternativa sería la siguiente:

1. Sabemos que entre 1 y 19 podemos tener entre 1 y 10 elementos en el vector de número impares.
2. Asignamos un valor inicial de 10 al localizador (suponemos 10 elementos).
3. Asignamos todos los impares que existan a elementos de la matriz existentes.
4. Conocidos los elementos válidos, redimensionamos la matriz eliminando los innecesarios.

El cambio en el pseudocódigo sería:

```
3. Impar = - 1 : i = 0
4. Redimensionar Valor(10)
5. Hacer
    i = i + 1
    Impar = Impar + 2
    Valor(i) = Impar
Repetir Mientras Impar < Dato
6. Redimensionar Valor(i)
7. Cantidad = i
```

Hemos sustituido la repetición de un proceso de redimensionar n veces por redimensionar dos veces, lo cual nos resulta ventajoso casi en cualquier circunstancia.

Si $i = 10$ Redimensionar Valor(i) resulta redundante pero no nos afecta al no crearse ni destruirse datos. Si se quiere evitar la redundancia se puede poner una condición de entrada Si $i < 10$.

¿Qué hubiera pasado si en vez de entre 1 y 19 se admitieran datos entre 1 y 1999? Antes de entrar a valorar el por qué de los extremos 1 y 19, pensemos en la asignación de contenido al vector Valor en el módulo Genera. Si el número de impares puede ser muy grande, mantener un redimensionamiento por bucle hará más lento y pesado todavía el proceso. Pero por otro lado ya no son 10 unidades el rango de variación posible del vector, sino 1000. Y crear un vector gigante para después usarlo mínimamente tampoco es demasiado eficiente. Es como contratar un gran camión container para al final... transportar en él una caja de zapatos.

Próxima entrega: CU00215A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60