

MÓDULO 1. INTRODUCCIÓN A LA PROGRAMACIÓN

Relación de Problemas N° 7 (Parte II) Arrays

Problemas.

8. La denominada *Criba de Eratóstenes* es un método para determinar los números primos entre 1 y N, siguiendo los siguientes pasos:
- Se escriben los números naturales entre 1 y N.
 - Se tacha el 1.
 - Se deja el 2 y se tachan todos los demás números pares.
 - Se deja el 3 y se tachan todos sus múltiplos.
 - Como el 4 ya está tachado, pasamos al 5, que se deja y se tachan todos sus múltiplos (los del 5).
 - Etc.

Así, cuando pasemos del 13, estarán tachados 14, 15 y 16, con lo que seguimos el proceso en el 17. El proceso acaba cuando llegamos a la raíz cuadrada de N. Los números que queden sin tachar, serán primos. La siguiente figura muestra este método aplicado del 1 al 100. Aparecen en blanco los números no marcados y que, por tanto, son primos.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28
29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76
77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99
100

Se pide crear un procedimiento denominado `eratostenes` que, mediante la criba descrita y haciendo uso de arrays, debe tomar como parámetro un natural N e imprimir por pantalla todos los números primos del 1 al N.

Crea también una función `main` para comprobar que el procedimiento se ha codificado correctamente.

9. Escriba un programa que efectúe la conversión de un número natural en base 10 a otra determinada base, sabiendo que el resultado no sobrepasará los 50 dígitos. El usuario introducirá primero el número en base 10 y después la base a la que convertirlo (el programa debe asegurarse de que la base no sea ni menor de 2 ni mayor de 9).
- Nota: Recordemos que para obtener la representación en una base b de un número en decimal, dividimos entre b primero el número y después sucesivamente los diferentes

cocientes que se vayan obteniendo hasta que el cociente sea cero. Los diferentes restos obtenidos en esas sucesivas divisiones constituyen la representación en dicha base b (pero en orden inverso a como se han ido calculando). Por ejemplo, para el número decimal 26 en base 2 es 11010.

$$\begin{array}{r}
 26 \mid 2 \\
 0 \ 13 \mid 2 \\
 1 \ 6 \mid 2 \\
 0 \ 3 \mid 2 \\
 1 \ 1 \mid 2 \\
 1 \ 0
 \end{array}$$

10. Dos colecciones de números enteros son iguales si contienen los mismos elementos y en el mismo orden relativo, suponiendo que el primer elemento sigue al último. Por ejemplo, estas dos colecciones son iguales:

1, 2, 3, 4, 5
3, 4, 5, 1, 2

Suponiendo que no hay elementos repetidos, diseña un algoritmo que lea de teclado dos colecciones de números enteros (el usuario indicará en primer lugar la cantidad de números que va a introducir por cada colección y después las colecciones) y determine si son iguales conforme al criterio establecido anteriormente.

11. Los alumnos de un curso de “Programación de Dispositivos Android” desean celebrar una comida un día del presente mes (marzo) en el que puedan acudir todos. Se pide realizar un algoritmo que recoja de cada alumno los días que le vendría bien ir a la comida, e imprima las fechas concordantes para todos los alumnos. Los datos se introducirán por teclado. Primero se introducirá el número de alumnos que intervienen. Después, por cada alumno se introducirá una única línea con los números de los días libres separados por espacios (un 0 para terminar).
12. Diseña una función que recibe como parámetro un array de valores enteros `val` y devuelve otro array de enteros `ind`, de forma que el contenido de cada una de sus celdas es un índice del array `val`. Los índices estarán almacenados de forma que si recorremos el array `ind` de izquierda a derecha, y visitamos las celdas de `val`, cuyo índice nos vamos encontrando en `ind`, los valores de `val` se recorrerán en orden creciente de menor a mayor. El array `val` no se podrá modificar, ni se puede hacer una copia del mismo.

Crea también una función `main` para comprobar que la función se ha codificado correctamente.

Por ejemplo:

0	1	2	3	4
10	5	-7	0	12

`val`

0	1	2	3	4
2	3	1	0	4

`ind`

13. Consideremos un vector V que contiene N valores naturales (array de naturales de tamaño N , siendo N una constante cualquiera). Definimos el centro c del vector V como el índice entre 1 y $N-2$ que verifica la siguiente propiedad:

$$\sum_{i=0}^{c-1} (c-i) * V[i] = \sum_{j=c+1}^{n-1} (j-c) * V[j]$$

Esta propiedad no siempre se verifica; en ese caso, decimos que el vector no tiene centro.

Diseña un algoritmo lea de teclado un vector V y muestre por pantalla el centro del mismo o bien una indicación de que no existe. A continuación se detallan dos ejemplos (suponemos que $N = 5$):

1er Ejemplo:

El contenido del vector es: 6 2 3 0 1

El centro de este vector es el índice 1 (casilla donde está el 2) ya que al calcular los sumatorios:

- Sumatorio izquierda: $(1-0)*V[0] = 1*6 = 6$

- Sumatorio derecha: $(2-1)*V[2]+(3-1)*V[3]+(4-1)*V[4] = 1*3+2*0+3*1 = 6$

2º Ejemplo:

El contenido del vector es: 1 2 1 1 0

Este vector no tiene centro