Práctico 2 – Resolución de algoritmos

OBJETIVOS

- Presentar metodologías avanzadas de programación y técnicas de resolución de problemas como partición y refinamiento.
- Presentar estándares y prácticas recomendadas de programación (especificación semántica y sintáctica).
- Repasar estructuras de control y poner énfasis en la programación detallada.

En todos los casos se deberá indicar las PRE/POS condiciones y realizar corridas a mano.

Ejercicio 1

Dado un número natural n, indicar si es primo o no. El nombre de la función debe ser “esPrimo” y el tipo del valor de retorno deber ser Boolean.

Nota: Un número es primo cuando sólo es divisible por sí mismo y por la unidad.

Ejercicio 2

Dado un número natural n, mostrar por pantalla el número primo más cercano que sea mayor al propio número n.

Ejercicio 3

Usando sólo sumas y restas, calcular y mostrar por pantalla el resto y el cociente de la división entera de dos números naturales a y b.

Ejercicio 4

Dados dos números enteros a y b, obtener su máximo común divisor. El nombre de la función debe ser “calcularMCD” y el tipo del valor de retorno deber ser Integer.

Nota: El MCD es el divisor más grande que ambos números tienen en común. Se sugiere aplicar el algoritmo de Euclides que se explica a continuación:

- Dados dos números n>m, comenzamos realizando la división entera de n entre m.
- Cada paso consiste en una nueva división, en la que el dividendo es el número que actuó de divisor en la división anterior y el divisor es el número que se obtuvo como resto en la división anterior.
- Cuando en una división se obtiene resto nulo, el máximo común divisor de los números de los que partimos será el número que ha actuado como divisor en esa última división efectuada y que resultó ser una división exacta.

Ejercicio 5
Diseñar una rutina que, dadas dos variables, numerador y denominador simplifique la fracción numerador / denominador (sugerencia usar el algoritmo de Euclides).

Ejercicio 6
Dado un número determinar si es capicúa. El nombre de la función debe ser “esCapicua” y el tipo del valor de retorno deber ser boolean.
Ejemplos de números capicúas: 24842, 11, 565, 1799971

Ejercicio 7
Listar por pantalla todos los números capicúas de 5 cifras (desde el 10001 en adelante).

Ejercicio 8
a) Escriba una función `int maximoDelVector (int v[ ], int desde, int hasta)` que devuelve el máximo valor encontrado en el vector v considerando únicamente las posiciones entre desde y hasta.

b) Escriba una función `int posicionDelMaximoDelVector (int v[ ], int desde, int hasta)` que devuelve la posición del máximo valor encontrado dentro del vector v considerando únicamente las posiciones entre desde y hasta.

Ejercicio 9
Escriba un procedimiento `void ordenar (int v[ ], int n)` que reciba un vector de n elementos y lo ordene.

Ejercicio 10
a) Escriba una función promedio que reciba un vector de N elementos y retorne el promedio.

b) Dado un vector de N elementos, hallar su mínimo, la cantidad de veces que se repite y la posición donde aparece por primera vez.

Ejercicio 11
Dados dos vectores ordenados v1 y v2 de dimensiones n1 y n2 respectivamente escribir una función que genere un tercer vector v3, también ordenado, que contenga los elementos de los dos anteriores.

Ejercicio 12
Escribir un procedimiento que reciba un vector de N enteros y dos variables, max1 y max2, y cargue en ella los dos números mayores del vector (discutir posibles post--condiciones).