

PRÁCTICAS DE LA ASIGNATURA MATEMÁTICA DISCRETA.
Ingenierías Técnicas de Informática de Gestión y de Sistemas.
2º Curso. Primer Cuatrimestre.
Curso 2006-2007

Tema 2: TEORÍA DE NÚMEROS.

1. Crea una función que reciba dos números a y b y devuelva su máximo común divisor. Utiliza para ello el algoritmo de Euclides. (Fácil)
2. Crea una función que reciba dos números a y b y calcule de modo recursivo su máximo común divisor. (Medio)
3. Crea una función que reciba dos números a y b y devuelva x e y tales que $mcd(a, b) = xa + yb$. (Medio)
4. Crea una función que reciba dos números a y b y devuelva x e y tales que $mcd(a, b) = xa + yb$, calculándolos de modo recursivo. (Difícil)
5. Crea una función que reciba dos números a y b y devuelva el mínimo común múltiplo. (Fácil)
6. Crea una función que reciba dos números a y b y devuelva *true* si son primos entre sí y *false* en caso contrario. (Fácil)
7. Crea una función que reciba tres números a , b y c y devuelva *true* si son primos entre sí y *false* en caso contrario. (Fácil)
8. Crea una función que reciba un vector con n números y devuelva *true* si son primos entre sí y *false* en caso contrario. (Medio)
9. Crea una función que reciba un número n y devuelva los números primos menores o iguales a n . Utiliza para ello la criba de Eratóstenes. (Medio)
10. Crea una función que reciba un número natural y devuelva *true* si el número es primo y *false* si no lo es. (Medio)
11. Crea una función que reciba un número natural n y devuelva dos números a, b tales que $a = 1$ y $b = n$ si el número es primo y $1 < a < b < n$ si n no es primo. (Medio)
12. Crea una función que reciba un número natural n y devuelva los números primos que lo dividen. Para calcular los números primos, se basará en la recursividad y utilizará la función creada en el apartado anterior. (Difícil)
13. Crea una función que reciba un número n y devuelva su descomposición en factores primos. Para devolver la descomposición en factores primos, se puede utilizar dos vectores (uno para los factores primos y otro para su multiplicidad) o una lista enlazada. (Difícil)

14. Crea una función que reciba dos números a y b , calcule la descomposición en factores primos de cada uno (llamando a alguna función que haga eso, por ejemplo, factores) y devuelva el máximo común divisor de a y b . (Fácil)
15. Crea una función que reciba dos números a y b , calcule la descomposición en factores primos de cada uno (llamando a alguna función que haga eso, por ejemplo, factores) y devuelva el mínimo común múltiplo de a y b . (Fácil)
16. Crea una función que reciba dos números a y b , calcule la descomposición en factores primos de cada uno (llamando a alguna función que haga eso, por ejemplo, factores) y devuelva *true* si a y b son primos entre sí y *0* en caso contrario (para ver si son primos, usará la descomposición en factores primos). (Fácil)
17. Crea una función que reciba dos números a y b , calcule la descomposición en factores primos de cada uno (llamando a alguna función que haga eso, por ejemplo, factores) y devuelva *true* si a es divisor de b y *false* en caso contrario. (Fácil)
18. Crea una función que reciba los coeficientes a y b , el término independiente c de la ecuación diofántica lineal $ax + by = d$ y un número n y devuelva *true* en caso de que tenga solución y *false* en caso de que no la tenga. Además, si tiene solución devolverá una solución. (Medio)
19. Crea una función que reciba un número n y devuelva las soluciones de la ecuación $x^2 - y^2 = n$. Puedes devolver las soluciones como dos vectores o como dos listas enlazadas. (Medio)
20. Crea una función que reciba un número n y devuelva las ternas pitagóricas tales que $0 < x, y, z < n$. Hazlo usando la ecuación $x^2 + y^2 = z^2$. Puedes usar las propiedades de las ternas, pero no la expresión de las soluciones. (Medio)
21. Crea una función que reciba un número n y devuelva las ternas primitivas pitagóricas tales que $0 < x, y, z < n$. Hazlo usando la expresión de las soluciones. Se valorará positivamente si la función devuelve todas las ternas pitagóricas en lugar de sólo las primitivas. (Medio)
22. Crea una función que reciba dos números enteros a, b, m , con $m > 0$ y devuelva las soluciones (módulo m) de $ax \equiv b \pmod{m}$. (Medio)
23. Crea una función que reciba cuatro números enteros b_1, b_2, m_1, m_2 , con m_1 y m_2 primos entre sí y devuelva la solución módulo $m_1 m_2$ de

$$\begin{cases} x \equiv b_1 \pmod{m_1} \\ x \equiv b_2 \pmod{m_2} \end{cases}$$

(Medio)

24. Crea una función que reciba seis números enteros $a_1, a_2, b_1, b_2, m_1, m_2$, con $\text{mcd}(a_1, m_1) = 1$, $\text{mcd}(a_2, m_2) = 1$, y $\text{mcd}(m_1, m_2) = 1$, y devuelva la solución módulo $m_1 m_2$ de

$$\begin{cases} a_1 x \equiv b_1 \pmod{m_1} \\ a_2 x \equiv b_2 \pmod{m_2} \end{cases}$$

(Medio)

25. Crea una función que reciba un número n y devuelva $\phi(n)$ (usa para ello la definición de $\phi(n)$).
(Fácil)
26. Crea una función que reciba n y calcule $\phi(n)$ a partir de la descomposición en factores primos de n y las propiedades de ϕ .
(Medio)
27. Crea una función que reciba tres enteros, a , n y m , con $\text{mcd}(a, m) = 1$, y devuelva el resto de dividir a^n entre m . Como si elevamos un entero a otro en general nos devuelve un entero que excede el tamaño máximo, usa para calcular el resto la función de Euler.
(Fácil)
28. Crea una función que reciba las cifras de un número en base m (un vector o una lista con números entre 0 y $m - 1$) y otra base n y devuelva el número en base n (puedes devolverlo como un vector o como una lista).
(Medio)
29. Crea una función que reciba m y n y devuelva el criterio de divisibilidad por m en base n . Almacena los números sólo hasta que aparezca un 0 o un 1.
(Fácil)
30. Crea una función que reciba un número a en base n y un vector v con el criterio de divisibilidad entre m (por ejemplo, si $n = 10$ y $m = 8$, entonces $v = (1, 2, 4, 0)$, si $n = 10$ y $m = 7$, $v = (1, 3, 2, 6, 4, 5, 1)$) y devuelva si es divisible entre m .
(Medio)

Normas de entrega:

1. Al finalizar la práctica, se entregarán únicamente los archivos fuentes, correspondientes a las funciones que se tenían que codificar y a un programa principal en el que se probaron las funciones.
2. En cada uno de los fuentes que se entreguen, se pondrá en la cabecera un comentario con el número del grupo y los nombres y apellidos de los integrantes.
3. Cada archivo debe estar convenientemente explicado mediante comentarios. Se valorará positivamente que se utilicen los resultados de la Teoría de Números para aumentar la eficiencia de los programas. En ese caso se debe incluir un comentario con el resultado que se haya utilizado.
4. Los archivos se guardarán en un directorio cuyo nombre será

Grupo + ngrupo + inicialesapellidos.

Por ejemplo, si vuestro grupo es el 12 y los apellidos son Álvarez, Fernández y Rodríguez, el directorio se llamará *grupo12afr*.

5. El directorio se comprimirá en un archivo con el mismo nombre y la extensión correspondiente al programa de compresión utilizada.
6. El archivo comprimido se enviará a *bravo.trinidad@gmail.es*, antes de la fecha indicada en la página web de la asignatura.