Problemas de programación de computadores

Autor: Prof. Jonatan Gomez Perdomo, Ph.D.

1. La granja

En una granja se crían un número de V - Vacas, A - Aves (pollos y gallinas) y E - escorpiones. Las vacas están encerradas en un corral de $N \times M$ metros cuadrados, las aves en un galpón y los escorpiones en vitrinas.

1. Si una vaca necesita M metros cuadrados de pasto para producir X litros de leche, ¿cuántos litros de leche se producen en la granja?.

2. Si 1/3 de las aves que hay en la granja son gallinas, y la mitad de las gallinas ponen 1 huevo cada 3 días y la otra mitad 1 huevo cada 5 días, ¿en un mes cuántos huevos producen? (1 mes ≡ 30 días).

3. Si los escorpiones de la granja se venden a China, y hay escorpiones de tres diferentes tamaños: pequeños (con un peso de 20 gramos), medianos (con un peso 30 gramos) y grandes (con un peso de 50 gramos), ¿cuántos kilos de escorpiones se pueden vender sin que decrezca la población a menos de 2/3?.

4. Al granjero se le daño el corral y no sabe si volver a cercar el corral con madera, alambre de púas o poner reja de metal. Si va a cercar con madera debe poner 4 hileras de tablas, con varilla 8 hileras y con alambre solo 5 hileras, él quiere saber que es lo menos costoso para cercar si sabe que el alambre de púas vale P por metro, las tablas a Q por metro y las varillas S por metro. Dado el tamaño del corral y los precios de los elementos, ¿cuál cerramiento es el más económico?.

2. Numéricos

5. Función potencia de un entero elevado a un entero.

6. Una función que determine si un número es divisible por otro.

7. Determinar si un número es primo.

8. Dados dos naturales, determinar si son primos relativos.

9. Determinar si un número es múltiplo de la suma de otros dos números.

10. Dados los coeficientes de un polinomio de grado dos, evaluar el polinomio en un valor dado.

11. Dados los coeficientes de un polinomio de grado dos, calcular coeficiente lineal de la derivada.

12. Dados los coeficientes de un polinomio de grado dos y un número real, evaluar la derivada del polinomio en ese número.

13. Dado un natural, determinar si es un número de Fibonacci o no.

3. Geométricos

14. Dadas la pendiente y el punto de corte de dos rectas, determinar si son paralelas, perpendiculares o ninguna de las anteriores.

15. Dadas la pendiente y el punto de corte de dos rectas, determinar los puntos de intersección al origen.

16. Dado el radio de un círculo, calcular el área del triángulo que circunscribe el circulo (triangulo afuera).

17. Dado el radio de un círculo, calcular el área y perímetro del cuadrado, pentágono y hexágono adentro (inscrito en un círculo) y afuera (inscribiendo al círculo).

18. Si una araña utiliza un patrón de hexágono regular para su telaraña, y cada hexágono está separado del otro por 1cm, y la araña quiere hacer una telaraña de πr^2, ¿qué cantidad de telaraña requiere la araña?.

4. Varios

19. Si en la UN están podando árboles y cada rama tiene P hojas, y a cada árbol le quitaron K ramas, cuántos árboles se deben podar para obtener T hojas?.

20. Si un amigo, no tan amigo, me presta K pesos a i pesos de interés diario, ¿cuánto le pagaré en una semana si el interés es simple?, ¿y cuánto si el interés es compuesto?.

21. Un niño se la pasó jugando con fichas de lego, tenia dos tipos de fichas de lego, fichas de cuadros de 1×1 (rojas) y fichas de cuadros de 2×1 (azules), y le dieron una base de $1 \times n$ cuadritos, ¿de cuántas formas distintas puede ubicar las fichas rojas y azules sobre la base?, ¿y si le dan una ficha amarilla de 1×3?.

5. Arreglos

22. Implementar la criba de Eratostenes para calcular los números primos en el rango 1 a n, donde n es un número natural dado por el usuario.

23. Desarrollar un algoritmo que calcule la suma de los elementos de un arreglo de números enteros (reales).

24. Desarrollar un algoritmo que calcule el promedio de un arreglo de enteros (reales).

25. Desarrollar un algoritmo que calcule el producto punto de dos arreglos de números enteros (reales) de igual tamaño. Sean $v = (v_1, v_2, \ldots, v_n)$ y $w = (w_1, w_2, \ldots, w_n)$ dos arreglos, el producto de v y w (notado $v \cdot w$) es el número: $v_1 * w_1 + v_2 * w_2 + \cdots + v_n * w_n$.

26. Desarrollar un algoritmo que calcule el mínimo de un arreglo de números enteros (reales).

27. Desarrollar un algoritmo que calcule el máximo de un arreglo de números enteros (reales).

28. Desarrollar un algoritmo que calcule el producto directo de dos arreglos de enteros (reales) de igual tamaño. Sean $v = (v_1, v_2, \ldots, v_n)$ y $w = (w_1, w_2, \ldots, w_n)$ dos arreglos, el producto directo de v y w (notado $v * w$) es el vector: $(v_1 * w_1, v_2 * w_2, \ldots, v_n * w_n)$.

29. Desarrollar un algoritmo que determine la mediana de un arreglo de enteros (reales). La mediana es el número que queda en la mitad del arreglo después de ser ordenado.

30. Hacer un algoritmo que deje al final de un arreglo de números todos los ceros que aparezcan en dicho arreglo.

    Ejemplo.

    vector original: $(1, 6, 0, 7, -3, 8, 0, -2, 11)$

    vector salida: $(1, 6, 7, -3, 8, -2, 11, 0, 0)$

    Ejemplo.

    vector original: $(0, 11, 36, 10, 0, 17, -23, 81, 0, 0, 12, 11, 0)$

    vector salida: $(11, 36, 10, 17, -23, 81, 12, 11, 0, 0, 0, 0, 0)$

31. Suponga que un arreglo de enteros esta lleno de unos y ceros y que el arreglo representa un número binario al revés. Hacer un algoritmo que calcule los números en decimal que representa dicho arreglo de unos y ceros.

    Ejemplo.

    Entrada: $(0, 1, 0, 1, 0, 1, 1)$ (representa el número 1101010).

    Salida: 106

    Ejemplo.

    Entrada: $(1, 0, 0, 1, 0, 1, 1, 1, 1)$ (representa el número 111101001).

    Salida: 389

32. Hacer un algoritmo que dado un número entero no negativo, cree un arreglo de unos y ceros que representa el número en binario al revés.

    Ejemplo.

    Número: 106

    Arreglo: $(0, 1, 0, 1, 0, 1, 1)$ (representa el número 1101010)

Ejemplo.

Número: 389

Arreglo: $(1,0,0,1,0,1,1,1,1)$ (representa el número 111101001)

33. Hacer un algoritmo que calcule el Máximo Común Divisor (MCD) para un arreglo de enteros positivos.

    Ejemplo.

    Arreglo: $(12,20,14,124,72,2458)$

    MCD del arreglo: 2

34. Hacer un algoritmo que calcule el Mínimo Común Múltiplo (MCM) para un arreglo de enteros positivos.

    Ejemplo.

    Arreglo: $(12,20,30,15)$

    MCD del arreglo: 60

6. Conjuntos como arreglos

Un arreglo de elementos de tipo $\mathbb{T}$ se puede utilizar para representar un conjunto finito de elementos del tipo $\mathbb{T}$. Esta representación es como sigue:

El conjunto $A = \{x_0, x_1, x_2, \ldots, x_{n-1}\}$ se representa como el arreglo $(x_0, x_1, x_2, \ldots, x_{n-1})$.

Usando esta representación hacer un programa que le permita al usuario leer dos conjuntos de enteros y escoger mediante un menú, una de las siguientes operaciones sobre ellos:

35. **Unión:** Calcula en un arreglo la unión de los conjuntos y la imprime.
36. **Intersección:** Calcula en un arreglo la intersección de los conjuntos y la imprime.
37. **Diferencia:** Calcula en un arreglo la diferencia del primero con el segundo y la imprime.
38. **Diferencia simétrica:** Calcula en un arreglo la diferencia simétrica de los conjuntos y la imprime.
39. **Pertenece:** Lee un entero y determina si el elemento pertenece o no a cada uno de los conjuntos y lo imprime.
40. **Contenido:** Determina sí el primer conjunto esta contenido en el segundo y lo imprime.
41. **Salir:** Permite al usuario salir de la aplicación.

Después de realizada la operación, el menú se debe presentar de nuevo hasta que el usuario desee salir.

Se debe verificar que el arreglo no tenga elementos repetidos.

Suponga ahora que los elementos de tipo $\mathbb{T}$ se encuentran ordenados totalmente. La representación anterior se puede modificar de tal manera que las operaciones anteriores sean implementadas de manera más eficiente. La idea es mantener el conjunto de manera ordenada.

42. Desarrollar el programa anterior usando la representación modificada con las operaciones entre conjuntos optimizadas.

7. Polinomios como arreglos

Un polinomio de grado n, como $P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x^1 + a_0 x^0$ se puede representar mediante un arreglo de reales de la siguiente manera: $(a_0, a_1, \ldots, a_{n-1}, a_n)$.

Usando esta representación hacer un programa que le permita al usuario leer dos polinomios y escoger mediante un menú, una de las siguientes operaciones sobre dichos polinomios:

43. **Evaluar:** Lee un real e imprime la evaluación de los dos polinomios en dicho dato.
44. **Sumar:** Calcula el polinomio suma y lo imprime.
45. **Resta:** Calcula el polinomio resta y lo imprime.
46. **Multiplicar:** Calcula el polinomio multiplicación y lo imprime.
47. **Dividir:** Calcula el polinomio división del primer polinomio por el segundo y lo imprime.
48. **Residuo:** Calcula el polinomio residuo de la división del primero por el segundo y lo imprime.
49. **Salir:** Permite salir de la aplicación al usuario.

Después de realizada la operación el menú se debe presentar de nuevo hasta que el usuario desee salir.

8. Matrices

50. Desarrollar un algoritmo que permita sumar dos matrices de números reales (enteros).
51. Desarrollar un algoritmo que permita multiplicar dos matrices de números reales (enteros).
52. Desarrollar un programa que sume los elementos de una columna dada de una matriz.
53. Desarrollar un programa que sume los elementos de una fila dada de una matriz.
54. Desarrollar un algoritmo que determine si una matriz es mágica. Se dice que una matriz cuadrada es mágica si la suma de cada una de sus filas, de cada una de sus columnas y de cada diagonal es igual.

Ejemplo.

8	1	6
3	5	7
4	9	2

Ejemplo.

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

55. Desarrollar un algoritmo que dado un entero, reemplace en una matriz todos los números mayores a un número dado por un uno y todos los menores o iguales por un cero.

    Ejemplo. Si el número dado es 5 y la matriz es

8	1	6
3	5	7
4	9	2

    Se debe modificar y dejar dicha matriz como:

1	0	1
0	0	1
0	1	0

56. Desarrollar un programa que calcule el determinante de una matriz cuadrada.

57. Desarrollar un programa que dadas una matriz cuadrada A y un arreglo de números reales del mismo tamaño B, calcule una solución x para el sistema de ecuaciones lineales $Ax = B$.

58. Desarrollar un programa que calcule la inversa de una matriz cuadrada.

59. Desarrollar un programa que tome un arreglo de tamaño n^2 y llene en espiral hacia adentro una matriz cuadrada de tamaño n.

    Ejemplo. Si el arreglo dado es: $(1, 2, 3, 4, 5, 6, 7, 8, 9)$ la matriz en espiral es:

1	2	3
8	9	4
7	6	5

9. Relaciones binarias como matrices

Una matriz se puede usar para representar una relación entre dos conjuntos A y B. Esta representación es como sigue: Si $A = \{x_0, x_1, x_2, \ldots, x_{n-1}\}$ y $B = \{y_0, y_1, y_2, \ldots, y_{m-1}\}$ una relación $\mathcal{R}$ de A en B se representa mediante una matriz de unos y ceros de tamaño $n \times m$, donde $A_{ij} = 1$ si el elemento x_i se relaciona con el elemento y_j, en caso contrario $A_{ij} = 0$.

Usando esta representación hacer un programa que le permita al usuario leer dos relaciones entre dos conjuntos y escoger mediante un menú, una de las siguientes operaciones sobre dichas relaciones:

60. **Unión:** Calcula e imprime la relación unión.
61. **Intersección:** Calcula e imprime la relación intersección.
62. **Simetría:** Determina si la primer relación es simétrica o no.
63. **Reflexividad:** Determina si la primer relación es reflexiva o no.
64. **Transitividad:** Determina si la primer relación leída es transitiva o no.
65. **Orden:** Determina si la primer relación leída es relación de orden o no.
66. **Equivalencia:** Determina si la primer relación leída es una relación de equivalencia.
67. **Función:** Determina si la relación es una función o no.
68. **Inyectividad:** Determina si la relación es una función inyectiva.
69. **Sobreyectividad:** Determina si la relación es una función sobreyectiva.
70. **Salir:** Permite al usuario salir del programa

Después de realizada la operación el menú se debe presentar de nuevo hasta que el usuario desee salir.

10. Cadenas

71. Desarrollar un algoritmo que reciba como entrada un carácter y de como salida el número de ocurrencias de dicho carácter en una cadena de caracteres.
72. Desarrollar un algoritmo que reciba como entrada dos cadenas y determine si la primera es subcadena de la segunda. (No se deben usar operaciones de subcadenas propias del lenguaje de programación).

    Ejemplos.

    - La cadena "`prosa`" es subcadena de la cadena "`la prosa debe ser armoniosa`".
    - La cadena "`pepito`" no es subcadena de la cadena "`el torpe pito de aire`".
    - La cadena "pe pito" si esta incluida en la cadena "`el torpe pito de aire`".

73. Desarrollar un algoritmo que reciba dos cadenas de caracteres y determine si la primera está incluida en la segunda. Se dice que una cadena está incluida en otra, si todos los caracteres (con repeticiones) de la cadena está en la segunda cadena sin tener en cuenta el orden de los caracteres.

    Ejemplos.

    - La cadena "`prosa`" está incluida en la cadena "`la profesora de ingles`".
    - La cadena "`pepito`" no esta incluida en la cadena "`un pedazo de tierra`", ya que le falta una "`p`".
    - La cadena "`pepito`" si esta incluida en la cadena "`tijeras o papel`".

74. Desarrollar un algoritmo que invierta una cadena de caracteres.

75. Desarrollar un algoritmo que determine si una cadena de caracteres es palíndrome. Una cadena se dice palíndrome si al invertirla es igual a ella misma.

    Ejemplos.

    - "`ala`" es palíndrome.
    - "`amor a roma`" es palíndrome.
    - "`al sur de Colombia`" NO es palíndrome.
    - "`anula las alas a la luna`" NO es palíndrome. (Al invertirla: "`anul al a sala sal aluna`") no es igual a la original.

76. Desarrollar un algoritmo que determina si una cadena de caracteres es frase palíndrome. Una cadena se dice frase palíndrome si la cadena al eliminarle los espacios es palíndrome.

    Ejemplos.

    - "`anula las alas a la luna`" es frase palíndrome.
    - "`amor a roma`" es frase palíndrome.
    - "`dabale arroz a la zorra el abad`" es frase palíndrome.

77. Desarrollar un algoritmo que realice el corrimiento circular a izquierda de una cadena de caracteres. El corrimiento circular a izquierda es pasar el primer carácter de una cadena como último carácter de la misma.

    Ejemplo.

    Entrada: "`al sur de Colombia`".

    Salida: "`l sur de Colombiaa`".

    Ejemplo.

    Entrada: "`Pepito va al colegio`".

    Salida: "`epito va al colegioP`".

78. Desarrollar un algoritmo que realice el corrimiento circular a derecha de una cadena de caracteres. El corrimiento circular a derecha de una cadena es poner el último carácter de la cadena como primer carácter de la misma.

    Ejemplo.

    Entrada: "`al sur de Colombia`".

    Salida: "`aal sur de Colombi`".

    Ejemplo.

    Entrada: "`Pepito va al colegio`".

    Salida: "`oPepito va al colegi`".

79. Desarrollar un algoritmo que codifique una cadena de caracteres mediante una cadena de correspondencias de caracteres dada. La cadena de correspondencias tiene como el primer carácter el carácter equivalente para el carácter '`a`', el segundo carácter para la '`b`'y así sucesivamente hasta la '`z`'. No se tiene traducción para las mayúsculas ni para la '`ñ`'.

    Ejemplo.

    Entrada: "`al sur de Colombia`" y "`qwertyuiopasdfghjklzxcvbnm`".

    Salida: "`qs lxk rt Cgsgdwoq`".

    Ejemplo.

    Entrada: "`al sur de Colombia`" y "`zxcvbnmasdfghjklqwertyuiop`".

    Salida: "`zg etw vb Ckgkhxsz`".

80. Desarrollar un algoritmo que decodifique una cadena de caracteres mediante una cadena de correspondencias de caracteres dada. La cadena de correspondencias tiene como el primer carácter el carácter equivalente para el carácter '`a`', el segundo carácter para la '`b`'y así sucesivamente hasta la '`z`'. No se tiene traducción para las mayúsculas ni para la '`ñ`'.

    Ejemplo.

    Entrada: "`qs lxk rt Cgsgdwoq`" y "`qwertyuiopasdfghjklzxcvbnm`".

    Salida: "`al sur de Colombia`".

    Ejemplo.

    Entrada: "`zg etw vb Ckgkhxsz`" y "`zxcvbnmasdfghjklqwertyuiop`".

    Salida: "`al sur de Colombia`".