Ayudantia 1

Carlos Lagos - carlos.lagosc@usm.cl

Ejercicios: Tipos de datos, operadores y estructuras de control

Visita a su Amigo

Un elefante ha decidido visitar a su amigo. La casa del elefante está en el punto 0, mientras que la casa de su amigo está en el punto x ($x > 0$) en la línea de coordenadas. En cada paso, el elefante puede moverse 1, 2, 3, 4 o 5 posiciones hacia adelante. Se busca determinar el número mínimo de pasos que necesita dar para llegar a la casa de su amigo.

Para resolver este problema, tu programa debe recibir el entero x e imprimir la cantidad mínima de pasos requeridos para que el elefante alcance la casa de su amigo.

Encontrar los divisores de un número

Desarrolla un programa que reciba un número entero x desde la consola. Este número estará en el rango de 1 a 10^5. Es decir, $1 \leq x \leq 10^5$ siempre se cumplirá. El programa debe mostrar todos los divisores de x en la consola.

Ejercicios: Arreglos

Subarreglo con suma máxima

Desarrolla un programa que reciba 10 enteros separados por espacios. El objetivo es imprimir la suma del subarreglo que tenga la suma máxima.

Utiliza una función.

Nota: Un subarreglo es una secuencia contigua de elementos de un arreglo. Por ejemplo, si el arreglo es [1, 2, 3, 4], entonces los subarreglos serían [1], [2], [3], [4], [1, 2], [2, 3], [3, 4], [1, 2, 3], [2, 3, 4], y [1, 2, 3, 4].

Ejercicios: Structs

Crear un vector de 2 dimensiones

Desarrolla un programa que reciba 2 enteros que representen las coordenadas (x, y) de un vector y los guarde utilizando una estructura (struct) que represente vectores.

Producto Punto entre Vectores

Crea una función que, dados dos vectores, calcule el producto punto entre ellos.

Definición del Producto Punto: El producto punto (también conocido como producto escalar) entre dos vectores $\vec{a}$ y $\vec{b}$ de dimensiones n se calcula como la suma de los productos de sus componentes correspondientes:

$$\vec{a} \cdot \vec{b} = \sum_{i=1}^{n} a_i \cdot b_i$$

Vectores Perpendiculares

Usando la función anterior, crea un programa que, dado dos vectores como entrada, verifique si son perpendiculares.

Para verificar si dos vectores $\vec{u}$ y $\vec{v}$ son perpendiculares, podemos utilizar el producto punto entre ellos. Según la ecuación:

$$\vec{u} \cdot \vec{v} = |\vec{u}| \cdot |\vec{v}| \cdot \cos\theta$$

donde θ es el ángulo entre los vectores. Si $\vec{u} \cdot \vec{v} = 0$, entonces los vectores son perpendiculares.

Por favor, guarda la definición del struct y las funciones para ejercicios que abordaremos más adelante.

Recursividad

Determinar si un Número es Par

Desarrolla una función recursiva que reciba un entero positivo x y determine si es par. Sin embargo, se debe tener en cuenta que no es posible utilizar el operador de módulo (%). Además, se desconoce cuáles son los números pares, pero se sabe que el 0 es par. Además, se establece la regla de que el antecesor de un número par es impar y el antecesor de un número impar es par.

Mínima Cantidad de Pasos para Alcanzar 0

Se te proporciona un número entero positivo n. En cada paso, puedes restar uno de los dígitos del número. ¿Cuál es la cantidad mínima de pasos necesarios para hacer que el número sea igual a 0?

Crea una función recursiva que reciba el número n y devuelva la cantidad mínima de pasos para llegar a 0.

Ejemplo: Para $n = 27$:
$27 \to 20 \to 18 \to 10 \to 9 \to 0$ (5 pasos)

Memoria dinámica y punteros

Eliminación de Vectores Perpendiculares

Tu tarea consiste en desarrollar una función que, dada una lista de vectores y un vector v, elimine de la lista aquellos vectores que sean perpendiculares a v.

Al eliminar un elemento, asegúrate de que los elementos restantes se desplacen para evitar espacios vacíos y que la memoria se gestione de manera eficiente. Es decir, si la lista tiene menos elementos después de la eliminación, asegúrate de que la memoria utilizada sea suficiente para almacenar los elementos restantes. Utiliza `new` y `delete` según sea necesario.

Por favor, guarda la definición del struct y las funciones para ejercicios que abordaremos más adelante.

Archivos

Filtrado de Vectores Perpendiculares

Se proporciona un archivo llamado "vectores.txt", que contiene vectores en el siguiente formato: la primera línea indica un entero n, que representa la cantidad de vectores en el archivo. Luego, las siguientes n líneas contienen la información de cada vector, donde cada línea contiene dos enteros x e y que representan el vector.

El objetivo es recibir un vector desde la consola, es decir, los valores de x e y. La tarea consiste en leer los vectores del archivo "vectores.txt" y eliminar aquellos que sean perpendiculares al vector ingresado por consola. Una vez filtrados los vectores, se deben guardar en el archivo "vectores_filtrados.dat" en formato binario, manteniendo la estructura de los vectores.