

Ejercicios básicos 4

Nota: La carga del vector debe ir en el main, la función solo debe calcular lo que pide el enunciado (salvo que se diga lo contrario explícitamente).
En todos los casos que se pida hacer una función, se debe hacer un main que lo valide en otro archivo (agregar el header correspondiente).

Ejercicio 1

Escribir un programa que pruebe la efectividad de la función de biblioteca rand(). Comience por inicializar a cero 100 contadores, como cuenta [0], cuenta [1], cuenta [2], ..., hasta cuenta [99]. Luego genere una gran cantidad de números pseudoaleatorios entre 0 y 99. Cada vez que ocurra un 0 se aumenta cuenta [0], cada vez que ocurra un 1 se aumenta cuenta [1], y así con todos los dígitos decimales. Por último imprima el número de ceros, unos, etc. que ocurrieron y el porcentaje de cada ocurrencia.
Se puede hacer todo en el main.

Ejercicio 2

Escribir una función que calcule el promedio de un vector.
El prototipo de esa función es:
float calcula_promedio (float vector [], int tam_vector);
Hacer el main que lo valide.

Ejercicio 3

Un tirador realiza 20 disparos a un blanco cuyo centro coincide con el origen de coordenadas. Leyendo la distancia al centro (el radio) de cada impacto, deberá calcularse el puntaje obtenido. El blanco tiene cuatro zonas que, de acuerdo a la distancia del impacto al origen (R), son:

ZONA

DENOMINACIÓN PUNTAJE

$0 \leq R \leq 1$ centro 10

$1 < R \leq 5$ medio 5

$5 < R \leq 10$ externo 1

$R > 10$ falla 0

El prototipo de la función que calcula el puntaje total es:

int calcula_puntaje (float radios [], int cantidad_de_disparos);

Ejercicio 4

Escribir una función que reciba un número **int** y separe cada dígito en un vector. Si el vector no tiene el tamaño necesario para guardar los dígitos, la función debe devolver un valor que indique dicho error.

El prototipo de la función es:

char separa_digitos (int numero, int digitos [], int tam_vector);

Ejercicio 5

Escribir una función que copie en orden inverso los valores de un vector (vector_original) en otro vector (vector_invertido).

El prototipo de la función es:

void invertir_vector (int vector_original [], int vector_invertido [], int tam_vector);

Ejercicio 6

Escribir **dos** funciones que copie de forma ordenada (una de forma ascendente y otra descendente) los valores de un vector (vector_original) en otro vector (vector_invertido).

Ejercicio 7

Escribir una función que calcule la mediana de un vector. Reutilizar funciones anteriores si es necesario.

Nota: La mediana representa el valor de la variable de posición central en un conjunto de datos ordenados.

Ej: vector [5] = {5, 1, 3, 2, 1}

vector_ordenado [5] = {1, 1, 2, 3, 5}

mediana = 2

Si el vector tiene una cantidad par de elementos, la mediana se calcula como la media aritmética entre los 2 valores centrales.

Ej: vector [6] = {5, 1, 3, 2, 1, 6}

vector_ordenado [6] = {1, 1, 2, 3, 5, 6}

mediana = $(2 + 3) / 2 = 2.5$