

Trabajo Práctico N° 3: Arreglos y Registros

Programación Avanzada-2025

Objetivo: Desarrollar un programa modular y eficiente que combine el uso de estructuras de datos (arreglos y registros) con estructuras de control (bucles y condicionales) y el diseño modular con funciones.

Consignas

1. Análisis de Defectos por Línea de Producción

Desarrolla una función que reciba un arreglo unidimensional de 10 enteros. Esta función debe solicitar y almacenar el número de productos defectuosos encontrados en cada una de las 10 líneas de producción. En la función principal, llama a esta función para cargar los datos y luego usa un bucle para encontrar y mostrar el número total de defectos reportados.

2. Validación de Número de Pedido

Crea una función booleana llamada validarPedido que reciba un número de pedido de 4 dígitos. Utiliza un bucle do-while para solicitar al usuario que ingrese un número de pedido y llama a la función validarPedido para verificar que el número cumpla con las siguientes condiciones: 1) que sea de 4 dígitos (entre 1000 y 9999), y 2) que sea un número par. El programa debe seguir pidiendo el número hasta que se cumplan ambas condiciones.

3. Registro de Tiempos de Ciclo

Crea una función registrarTiempos que reciba un arreglo unidimensional de 8 enteros por referencia. La función debe llenar el arreglo con el tiempo de ciclo (en segundos) de una operación, pidiendo el dato para cada una de las 8 horas de un turno. Luego, en la función principal, usa los datos del arreglo para calcular y mostrar el tiempo de ciclo promedio del turno.

4. Inventario de Materia Prima por Proveedor

Implementa un programa modular para gestionar un inventario bidimensional. Crea una función cargarInventario que reciba una matriz de 5x3 (int) por referencia, donde las filas representan 5 tipos de materia prima y las columnas a 3 proveedores diferentes. La función debe llenar esta matriz con los datos de stock. En el programa principal, muestra la matriz completa en formato de tabla.

5. Simulación de Inspección de Cajas (void)

Crea una función llamada simularInspeccion que reciba una matriz de 4x6 de caracteres. La función debe simular la inspección de 4 contenedores (C1 a C4), cada uno con 6 cajas. Llena la matriz con valores predefinidos: 'A' (Aprobado) o 'R' (Rechazado). Luego, usa un bucle anidado para recorrer la matriz y mostrar cuántas cajas en total fueron aprobadas y cuántas fueron rechazadas. La función no debe devolver valores.

6. Ficha de Equipos de Fábrica

Define una estructura Equipo con los campos: código (int), nombre (string) y estadoOperativo (bool). Crea una función cargarEquipo que reciba una variable

Equipo por referencia y pida al usuario que ingrese los datos. Luego, crea otra función evaluarEstado que reciba una variable Equipo y, usando un if-else, muestre un mensaje de "Operando" si el estado es true o "En Mantenimiento" si es false.

7. **Actualización de Stock en Almacén**

Define una estructura Producto con los campos codigo (int) y stock (int). Crea una función llamada actualizarStock que reciba un Producto por referencia. Dentro de la función, pregunta al usuario cuántas unidades se recibieron o se enviaron y modifica el campo stock de la variable. El programa principal debe mostrar el stock del producto antes y después de la llamada a la función para demostrar que se actualizó permanentemente.

8. **Gestión de Lotes de Producción**

Define una estructura Lote que contenga un numeroLote (int), una fechaProduccion (string) y la cantidad de defectos (int). Crea un arreglo de 5 Lote (arreglo de estructuras). Crea una función cargarLotes para llenar el arreglo con la información de los 5 lotes. Luego, crea una función encontrarLoteMayorDefectos que recorra el arreglo y devuelva el numeroLote que tuvo la mayor cantidad de defectos.

9. **Análisis de Calidad de un Lote**

Define una estructura LoteCalidad con los campos codigoLote (int) y un arreglo de 10 bool llamado inspeccionesDiarias. Cada true en el arreglo representa un día sin defectos. Crea una función cargarInspecciones que reciba un LoteCalidad por referencia y llene el arreglo. Luego, crea una función evaluarLote que reciba la estructura y devuelva un double que represente el porcentaje de días sin defectos. El programa principal debe mostrar el porcentaje para el lote analizado.

10. **Sistema de Control de Producción Completo**

Define una estructura ReporteProduccion con los siguientes miembros: un codigo (int), y una matriz unidades de 3x5 (int), donde las filas son turnos y las columnas son líneas de producción.

- Crea una función cargarReporte que reciba una variable ReporteProduccion por referencia. En esta función, usa un bucle anidado para solicitar las unidades producidas en cada turno y línea.
- Crea una función calcularPromedioTurno que reciba el reporte, el número de turno y devuelva el promedio de unidades producidas en ese turno.
- En la función principal, usa un bucle para iterar a través de los tres turnos, llamando a calcularPromedioTurno para cada uno y mostrando el resultado.
- Además, muestra el promedio de la producción total del día.