

ALGORITMOS Y ESTRUCTURAS DE DATOS

2023/1

PROFESOR: Manuel Alejandro Moscoso Domínguez
manuel.moscoso.d@gmail.com

Laboratorio Semana 2

En esta oportunidad realizaremos actividades que refuercen el trabajo con arreglos y sus funciones asociadas en C y además de revisar la gestión de un repositorio a través de GIT.

Objetivos

- Resolver ejercicios que involucren el uso de arreglos en C.
- Desarrollar algoritmos que permitan entregar una solución a los problemas entregados.

Ejercicios

Nomenclatura para nombre de archivos fuentes

El nombre del archivo en el cual se almacena el código fuente debe considerar el siguiente formato: labsemanaX_ejercicioY.EXT donde; X corresponde a la semana, la Y al número de ejercicios y EXT a la extensión del lenguaje de programación utilizado.

Ejercicio número 1

Tiene la tarea de crear un programa en C que ayude a realizar un seguimiento de los elementos recopilados por un jugador en un videojuego. El jugador puede recolectar hasta 100 tipos diferentes de elementos, cada uno con su propio número de identificación único.

El programa debe permitir al usuario ingresar los números de identificación de hasta 100 elementos recolectados, y luego debe mostrar un resumen de cuántos de cada elemento ha recolectado el jugador.

Ejercicio número 2

En el videojuego debe existir el concepto de intercambio de objetos para lo cual es necesario incluir algunas funcionalidades a lo ya realizado. Cada jugador tiene un inventario de artículos y nuestra tarea es escribir una función en C que fusione los inventarios de jugadores, creando un nuevo inventario que contenga las cantidades combinadas de cada artículo.

Todo esto es con el propósito de no exceder el límite que cada ítem tiene dentro del juego.

Ejercicio número 3

Parte importante del inventario y que los jugadores puedan intercambiar objetos es la posibilidad de poder agregar y eliminar elementos recopilados por los jugadores. Para esto es necesaria la creación de las funciones de agregar y eliminar elementos. El detalle específico debe considerar:

la función agregar: Si el tipo de artículo ya está en el inventario, la función debe aumentar la cantidad de ese artículo en uno. De lo contrario, debe agregar el artículo al inventario con una cantidad de uno. La función debe devolver el arreglo modificado.

La función eliminar: Si el tipo de artículo está en el inventario y su cantidad es mayor que uno, la función debería disminuir la cantidad de ese artículo en uno. De lo contrario, debería eliminar el artículo del inventario por completo. La función debe devolver el arreglo modificado.

Ejercicio número 4

Es necesario realizar el desarrollo de un programa que nos permita realizar un seguimiento del estado de salud diario de los pacientes durante la pandemia de COVID-19. El programa debe ser desarrollado en C debido a que será cargado en los monitores que se encuentra vigente en los hospitales y debe considerar las siguientes limitaciones:

- Desarrollo en C.
- Solo permite un máximo de 100 pacientes debido a que existen 100 espacios o habitación disponibles.
- Puede incluir un nombre o número de habitación y edad.
- La temperatura debe ser controlada como mínimo 7 días desde su ingreso de forma consecutiva.
- La fiebre se establece a partir de 37.5° C.
- Debe permitir la entrega de datos en grados C° o F°.

Es fundamental que realice el cálculo y pueda mostrar la temperatura promedio de cada paciente durante la última semana e identificar a cualquier paciente que haya tenido fiebre durante tres o más días consecutivos.

Ayuda memoria

```
#include <stdio.h>

//Definición de variables globales

#define MAX 10    // numero maximo.
#define umbral 7  // numero umbral

void funcionImprimir(int arreglo[]){
    for (int i = 0; i < MAX ; i++){
        printf("%d ",arreglo[i]);
    }
    printf("\n");
}

int* funcion(int arreglo[],int valor){
    if ( valor < umbral ) {
        arreglo[0] = valor;
    }
    return arreglo;
}

int main() {
    int miArreglo[MAX] = {0};
    funcionImprimir(miArreglo);
    funcion(miArreglo,8);
    funcionImprimir(miArreglo);
    return 0;
}
```