

TÉCNICAS DE PROGRAMACIÓN
CUARTA PRÁCTICA CALIFICADA
SEMESTRE ACADÉMICO 2021-1

Horario: Todos

Duración: 110 minutos

Elaborado por los profesores del curso.

ADVERTENCIAS:

- SE LES RECUERDA QUE, DE ACUERDO AL REGLAMENTO DISCIPLINARIO DE NUESTRA INSTITUCIÓN, CONSTITUYE UNA FALTA GRAVE COPIAR DEL TRABAJO REALIZADO POR OTRA PERSONA O COMETER PLAGIO. ESTO, Y EL HECHO DE ENCONTRAR CUALQUIER ARCHIVO YA SEA .c O .h CON FECHA U HORA DE CREACIÓN ANTERIOR A LA PRÁCTICA SERÁ CONSIDERADO UNA FALTA DE PROBIDAD Y POR LO TANTO AMERITARÁ LA ANULACIÓN DE LA PRUEBA.

INDICACIONES:

- LAS SOLUCIONES DEBERÁN DESARROLLARSE BAJO UN Estricto DISEÑO DESCENDENTE, por lo que NO SE CALIFICARÁN aquellos módulos que son llamados por otros que estén incompletos. Cada módulo no debe sobrepasar las 30 líneas de código aproximadamente.
- NO SE PUEDEN EMPLEAR ARCHIVOS DE DATOS AUXILIARES NI VARIABLES GLOBALES.
- En la calificación se tomará en cuenta el buen uso de los nombres de los identificadores, y el eficaz uso de comentarios.
- DEBE COLOCAR SU NOMBRE Y CÓDIGO EN CADA UNO DE LOS ARCHIVOS .h Y .c QUE EMPLEE EN SUS PROYECTOS, DE LO CONTRARIO SE LE DESCONTARÁ 0.5 PUNTOS POR CADA OMISIÓN. NO SE HARÁN EXCEPCIONES.

Cree un proyecto en NetBeans con el nombre: **“Semestre2021-1-Practica04”** (de no respetar este nombre se le descontarán dos (2) puntos de su nota final – NO SE HARÁN EXCEPCIONES) y en él desarrolle el programa que resuelva el problema que se describe a continuación.

En una empresa de transporte interprovincial de pasajeros se requiere de una aplicación en Lenguaje C para controlar su flota de buses, para esto se cuenta con dos archivos CSV como los que se muestran a continuación:

Buses.csv
M0G-547,López Reyes Juan Carlos, Ica,Arequipa,Moquegua,Tacna,25
A5R-175,Garcés Carpio Luciana,Huaraz,Chiclayo,20
...

El archivo contiene las rutas que han sido asignadas a los diferentes buses de la compañía. En cada línea aparece primero la placa del vehículo, le sigue el nombre del chofer de la unidad, las ciudades a las que llegará y en las que se detendrá para dejar pasajeros, y finalmente, viene la cantidad máxima de pasajeros que puede transportar la unidad (asientos del vehículo). Cada bus podrá pasar por un número diferente de ciudades, también se debe tomar en cuenta que varios buses pueden pasar por la misma ciudad.

Pasajeros.csv
78346512,Martel/Suarez/Patricia Susana, Ayacucho
22990375,Quispe/Torres/José, Moquegua
...

El archivo contiene una lista de personas que solicitaron un boleto para viajar a una ciudad del interior del país. En cada línea aparecerá el DNI de la persona, su nombre y la ciudad de destino.

Para realizar esta labor, la aplicación deberá cumplir los siguientes requerimientos:

- Definir las siguientes estructuras:
Bus.- con los siguientes campos: **placa** (cadena de caracteres), **chofer** (cadena de caracteres) y **numeroDeAsientos** (valor entero), **ciudadesDestino** (arreglo de cadenas de caracteres), **numeroDeCiudades** (valor entero), **pasajeros** (arreglo dinámico de estructuras tipo Pasajero, tenga en cuenta el número de asientos para el tamaño de este arreglo), **numeroDePasajeros** (valor entero). Tenga en cuenta que un bus llegará como máximo a 10 ciudades.
Pasajero.- con los siguientes campos: **dni** (valor entero), **nombre** (cadena de caracteres) y **destino** (cadena de caracteres). Los campos que manejen cadenas de caracteres deben definirse como **char***, esto es, como punteros, no como arreglos estáticos.

- Definir un arreglo de tipo **Bus** y cargarlo con la información de del archivo **buses.csv**, el archivo debe quedar ordenado descendientemente por el número de ciudades por las que pasará y en segundo lugar ascendientemente por el número de asientos.
- Leer los datos del archivo **pasajeros.csv**. Cada pasajero deberá ser asignado al primer bus del arreglo de buses que pase por su ciudad de destino. Si el bus ya no cuenta con asientos disponibles deberá buscar el siguiente bus, si ya no hay buses que pasen por la ciudad de destino, deberá colocar al pasajero en un arreglo auxiliar denominado **listaDeEspera**, de tipo **Pasajero**.

Confeccionar el siguiente reporte:

```

=====
Lista de pasajeros
=====
Vehículo No. 1
Placa: P2Q-123
Chofer: Ramirez Villa Jose Federico
Ruta: Ica - Arequipa - Moquegua - Tacna
Pasajeros:
Ica:
No.      DNI              Nombre
1        73556210         Pérez Roca Juan Carlos
2        56743892         Becerra Chavez Eva Maria Luisa
...
10       ...              ...
...
Arequipa
...
=====
Vehículo No. 2
Placa: ...
...
=====
PASAJEROS EN LISTA DE ESPERA
=====
No.      DNI              Nombre              Destino
1        10293847         Castillo Correa Marcos      Cusco
2        13579753         Salazar Quispe Andrés Gustavo  Huancavelica
3        ...              ...                          ...

```

Consideraciones:

- Los archivos solo se pueden leer una vez.
- Toda operación de búsqueda debe realizarse en una función independiente. No se considerará en la calificación los procesos de búsqueda que estén contenidos en el código de otro proceso. Las funciones de búsqueda deben considerar la posibilidad que el dato buscado no se encuentre.
- Se les proporcionará la biblioteca **funcionesCadena** (.h y .c) para que la incorporen en su proyecto. **Los archivos son propiedad intelectual de los profesores del curso por lo que no puede modificarlo, ni agregar o quitar algo a su contenido.**

ADVERTENCIAS:

- Obligatoriamente debe desarrollar su proyecto bajo NetBeans en Windows, no podrá desarrollarlo empleando otro IDE ni otro sistema operativo.
- Al finalizar la práctica, comprima la carpeta **Semestre2021-1-Practica04** en un archivo de tipo .zip (Solo debe usar el programa Zip que viene por defecto en el Windows, **NO deberá usar Win-RAR, RAR, 7Zip, etc.**) y súbalo a la tarea programada en Paideia para esta práctica. El nombre del archivo compilado será el mismo que el del proyecto.

CRITERIOS DE CALIFICACIÓN:

1. Si el programa entregado presenta más de tres errores de sintaxis serán calificados sobre la mitad del puntaje.
2. Si el programa no muestra los resultados o los muestren y no sean correctos, no podrán tener más del 75% de la nota.
3. Se descontará 15% de la nota si el programa define variables con nombres que no tengan sentido. Las variables deben empezar con una minúscula, se emplearán mayúsculas para separar las palabras compuestas (p. e.: baseInf).
4. Se descontará 15% de la nota si no se colocan comentarios relevantes, incluyendo un encabezado al inicio del programa en el que se indique el nombre del autor, la fecha, y una descripción de lo que hace programa).
5. No se calificará el código puesto como comentario.
6. No se calificarán aquellas funciones implementadas en el archivo main.c

San Miguel, 28 de junio del 2021