Los Proyectos de Software con los que trabajamos en el TP anterior  ahora se encuentran en un archivo de texto [proyectos.csv](https://uv.frc.utn.edu.ar/pluginfile.php/643780/mod_assign/intro/proyectos.zip), (hacer click en el nombre del archivo para descargar el archivo comprimido que lo contiene), en el que cada dato se separa del siguientes por caracteres *pipe* (“|”).  La primera línea del archivo contiene los encabezados, y debe ser salteada al procesar el archivo. Cada línea del archivo se separa de la siguiente con un salto de línea, y cada una tiene el siguiente formato:

* ***nombre\_usuario***: Cadena de caracteres que representa el nombre de usuario que subió el proyecto.
* ***repositorio***: Cadena de caracteres que representa el nombre del repositorio donde fue subido.
* ***descripcion***: Descripción en formato cadena de caracteres
* ***fecha\_actualizacion***: Cadena de caracteres que representa la  fecha de última actualización. Tiene el formato aaaa-mm-dd en dónde hay 4 dígitos para el año, dos del mes y dos del día separados por guiones.
* ***lenguaje***: Cadena de caracteres que representa el lenguaje de programación del proyecto.
* ***likes***: Cadena de caracteres que tiene un valor numérico que representa *miles de me gusta* indicados sobre el mismo.
* ***tags***: Es una lista de palabras separadas por comas.
* ***url***: Cadena de caracteres que representa la URL en la que está almacenado el proyecto.

**ACLARACIÓN MUY IMPORTANTE: Cuando abran el archivo, usen la siguiente variante de la función *open()* para evitar posibles fallas en la lectura debido al sistema de codificación de caracteres que se esté usando en cada plataforma:**

**m = open("libros.csv", mode="rt", encoding="utf8")**

Se pide desarrollar un programa con menú para las siguientes opciones:

**1.) Cargar:**Cargar el contenido del archivo en un *vector de registros* de proyectos (defina la clase Proyecto), donde cada registro tenga los siguientes campos:

* + **nombre\_usuario**: cadena de caracteres.
  + **repositorio**: cadena de caracteres.
  + **fecha\_actualizacion**: puede optar por almacenarla como una cadena de caracteres o como una variable de tipo registro Fecha que tenga los campos dia, mes y año. Tener en cuenta que si opta por un nuevo tipo de registro, cada vez que se muestre la fecha deberá mostrarse como una cadena de caracteres con el formato del archivo.
  + **lenguaje**: cadena de caracteres.
  + **likes**: Se debe almacenar como un valor numérico con punto flotante representando los miles de me gusta. Por ejemplo 69.7k debe almacenarse como 69.7.
  + **tags**: vector de cadenas de caracteres. Para obtener el vector de tags se debe separar la cadena que se encuentra en la columna correspondiente del archivo de texto por comas. No todos los registros tienen tags.
  + **url**: cadena de caracteres.

Deben considerarse las siguientes reglas al generar el vector a partir del archivo de texto:

* La primera línea del archivo contiene el nombre de los campos y por lo tanto no debe ser tenida en cuenta.
* El vector debe mantenerse siempre ordenado por el campo repositorio. Se considerará incorrecta la solución de cargar todos los registros y ordenarlos al final.
* Las líneas que tengan el lenguaje en blanco no deben ser procesadas.

Una vez finalizada la carga del vector, mostrar la cantidad de registros que se cargaron y la cantidad de registros que se omitieron.

**2.) Filtrar por tag:**Cargar por teclado un tag (cadena de caracteres) y a partir del vector de proyectos, mostrar todos aquellos registros que contengan al tag en alguno de los elementos del vector alojado en el campo *tag*. Los proyectos deben mostrarse a razón de un registro por línea mostrando solamente el nombre del repositorio, la fecha de actualización y  cantidad de estrellas.

Las estrellas representan un rango de *miles de likes* del archivo:

* 1: de 0 a 10 k
* 2: de 10.1 a 20 k
* 3: de 20.1 a 30 k
* 4: de 30.1 a 40 k
* 5: mayor a 40 k

Se debe dar al usuario la opción de almacenar el listado en un archivo de texto separado por pipes donde se escriba una línea inicial con el nombre de las columnas (o campos) y luego una línea por cada registro de proyecto con todos sus campos (no solo los campos que se mostraron en el listado).

**3.) Lenguajes:**A partir del vector determinar la cantidad de proyectos por cada lenguaje de programación. Mostrar los lenguajes de programación y su cantidad ordenados de mayor a menor por cantidad.

**4.) Popularidad*:***Se quiere conocer los meses en los que se actualizan los proyectos, de acuerdo a la cantidad de estrellas. Para ello se pide***,***a partir del vector, generar una matriz donde cada fila sea un mes de actualización (no importa de qué año corresponde)  y cada columna una cantidad de estrellas. Cada celda deberá contener la cantidad de proyectos que tengan ese mes de actualización y esa cantidad de estrellas. Las estrellas representan los rangos de likes indicados en el punto 2.

Mostrar la matriz resultante como una tabla de filas y columnas. Indique, además, cuál es el total de proyectos actualizados en el mes m, siendo m un mes que se ingresa por teclado (valor entero del 1 al 12).

**5.) Buscar proyecto actualizado:** A partir del vector, buscar un repositorio con el nombre rep, siendo rep  un valor ingresado por teclado. Si existe mostrar sus datos y permitir reemplazar la URL del proyecto que se ingrese por teclado  y cambiar la fecha de actualización por la fecha actual (la fecha no se carga por teclado, investigue la manera de obtener la fecha actual y formatearla de igual manera en la que se encuentra en el archivo. Si no existe indicar con un mensaje de error.

**6.) Guardar populares:**A partir de la matriz generada en el punto 4, almacenar su contenido (sólo los elementos mayores a cero) en un archivo binario en el que cada elemento sea un registro en el que se representen los campos:

* mes del año.
* estrellas (El rango indicado en el punto 2).
* cantidad de proyectos.

**7.) Mostrar archivo:** Leer el contenido del archivo binario y volver a generar la matriz. Mostrarla en formato de tabla.

**8.) Salir:**Finaliza la ejecución del programa.