Algoritmos y Estructuras de Datos
Examen Final 03/12/2020 – Regulares 2015+ [Python y PE]

Un centro de investigación biomédico necesita un programa que le permita operar con los diferentes medicamentos que tiene bajo estudio. De cada **Medicamento**, se tiene una clave de identificación (una cadena que puede tener dígitos y caracteres), la descripción del alcance y efectos de ese medicamento (una cadena con un texto terminado en punto y con palabras separadas por un blanco. Por ejemplo: "Posible vacuna contra el maldito COVID-19.", "Propuesta en fase 3 para un tratamiento contra la leucemia.", etc.), el nombre del director del proyecto, el monto invertido en el proyecto, un número entre 1 y 30 que indica el tipo de medicamento (por ejemplo: 1: Cardíaco, 2: Inmunológico, etc.), y otro número pero entre 0 y 9 para indicar el tipo de avales que tiene el proyecto (por ejemplo: 0: Gubernamental, 1: Internacional, etc.).

En base a lo anterior, desarrollar un programa completo que disponga al menos de dos módulos:

- En uno de ellos, definir la clase Medicamento que represente al registro a usar en el programa, y las funciones básicas para operar con registros de ese tipo.
- En otro módulo, incluir el programa principal y las funciones generales que sean necesarias. Aplique las validaciones que considere necesarias. El programa debe basarse en un menú de opciones para desarrollar las siguientes tareas:

1. Generar un arreglo de registros que contenga los datos de todos los medicamentos en estudio. Puede generarlo en forma manual o aleatoria. El arreglo debe generarse de forma que en todo momento quede ordenado por clave de identificación, de menor a mayor. Debe considerar que esta opción puede ser invocada varias veces a lo largo del programa, y que en cada ejecución pueden agregarse tantos registros que desee el operador, sin eliminar los datos que ya estaban cargados. Se evaluará la eficiencia del algoritmo/estrategia que aplique para cumplir este punto.
2. Muestre el arreglo generado, a razón de un registro por línea en la consola de salida.
3. Determine si existe en el arreglo un medicamento en el que el nombre del director coincida con el valor **nom** que se carga por teclado. Si existe, retorne la cadena contenida en el campo **descripción** y detenga la búsqueda. Si no existe, retorne una cadena de la forma **'No existe'**. En ambos casos, muestre la cadena retornada.
4. Determine si existe en el arreglo un medicamento en el que la clave de identificación coincida con el valor **k** que se carga por teclado. Si existe, muestre sus datos completos y detenga la búsqueda. Si no existe, agregue un nuevo registro con esa clave al vector, generando el resto de sus datos en forma aleatoria. Recuerde que el vector debe permanecer ordenado en todo momento según el valor de las claves de identificación.
5. Determine el monto acumulado que se ha invertido para cada uno de los posibles tipos de proyectos y por cada tipo posible de aval (un total de 30 * 10 = 300 acumuladores). Muestre sólo los resultados que sean mayores a un valor **m** que se carga por teclado.
6. Genere un **archivo binario** con todos los registros del vector en los que la longitud del campo **descripción** sea menor o igual a 25 y cuyo **tipo de medicamento** no sea el 10 ni el 15.
7. Muestre el archivo generado en el punto anterior, a razón de un registro por línea en la pantalla. Pero muestre solo los registros cuyo monto invertido sea mayor a 100000.
8. Tome la cadena retornada en el punto 3, y determine cuántas palabras de esa cadena contenían solo letras mayúsculas y al menos un dígito. Puede considerar que la cadena termina siempre con un punto, y que las palabras se separan entre ellas con un (y solo un) espacio en blanco. La cadena debe ser procesada caracter a caracter, a razón de uno por cada vuelta del ciclo que itere sobre ella.

Criterios generales de evaluación:

a.) Desarrollo del programa completo, incluyendo los dos módulos pedidos como mínimo, el menú correctamente planteado, funciones correctamente diseñadas y parametrizadas (cuando sea apropiado), validaciones cuando sean aplicables y estilo de código fuente: **[máximo: 2 puntos (8.3% del puntaje)]**

b.) Desarrollo correcto de los ítems 1 y 2: **[máximo: 4 puntos (16.6% del puntaje)]**

c.) Desarrollo correcto del ítem 3: **[máximo: 4 puntos (16.7% del puntaje)]**

d.) Desarrollo correcto del ítem 4: **[máximo: 4 puntos (16.7% del puntaje)]**

e.) Desarrollo correcto del ítem 5: **[máximo: 2 puntos (8.3% del puntaje)]**

f.) Desarrollo correcto de los ítems 6 y 7: **[máximo: 4 puntos (16.7% del puntaje)]**

g.) Desarrollo correcto del ítem 8: **[máximo: 4 puntos (16.7% del puntaje)]**

Para aprobar el parcial, el alumno debe llegar a un *total acumulado de al menos 60% del puntaje (es decir, alrededor de 14.4 puntos acumulados)*, pero obligatoriamente debe estar desarrollado el programa, funcionando y operativo.

NOTA	PORCENTAJE	CALIFICACIÓN
1		Insuficiente
2		Insuficiente
3		Insuficiente
4		Insuficiente
5		Insuficiente
6	60% a 68%	Aprobado
7	69% a 77%	Bueno
8	78% a 86%	Muy Bueno
9	87% a 95%	Distinguido
10	96% a 100%	Sobresaliente