

Fundamentos de Programación 2021. 1er Parcial. 21/05/2021. Tema A

Ej.1 (40 pts) Un desarrollador de software quiere analizar datos relacionados a las cantidades de descargas de uno de sus programas. El programa está disponible para 5 sistemas operativos (1-Android, 2-iOS, 3- Windows, 4-Linux, 5-macOS).

a) Escriba un programa que permita leer las cantidades de descargas de cada sistema operativo por cada mes del año. La carga de datos se debe realizar ingresando las cantidades ordenadas por mes: primero las 12 cantidades de Android, luego las 12 de Windows, etc).

El programa luego debe informar:

b) El promedio de descargas de cada mes (sumando todos los sistemas)

c) El sistema que más descargas tuvo en el año completo (**sistema_mas_elegido**), y cuántas fueron (**cant_descargas**).

d) En cuántos, y cuáles, meses las descargas para sistemas de escritorio (sistemas 3, 4 y 5) superaron a las descargas para sistemas móviles (sistemas 1 y 2).

 **Ejemplo de resolución:** https://youtu.be/EEEnTkR_Q4Jo?t=187

Ej.2 (35 pts) Un estudio contable lleva información de un conjunto de **N** monotributistas. Cada uno tiene un límite de facturación de acuerdo a su categoría. Si se pasa de ese límite (si sus facturas suman más que ese monto), debe "recategorizarse" (cambiar a otra categoría con un límite diferente).

Se requiere un programa que:

a) Permita ingresar el límite de facturación de cada monotributista (N montos, uno por cliente).

b) Permita cargar las facturas de un período. Por cada una se ingresan 2 datos: el nro de cliente (de 1 a N) y el monto de la factura. Se desconoce a priori la cantidad, por lo que estos datos finalizan cuando se ingresa el nro de cliente 0.

c) Informe cuáles clientes deben recategorizarse (que la suma de sus facturas ingresadas en el punto b, supera el límite ingresado en el punto a).

 **Ejemplo de resolución:** https://youtu.be/EEEnTkR_Q4Jo?t=3162

Ej.3 (35 pts) Una concesionaria que vende cierta marca de autos tiene más demanda que oferta. Los clientes que quieren comprar un auto quedan registrados en una lista de espera. Al anotarse en la lista, el cliente indica el modelo en el que está interesado y elige 2 colores entre los disponibles para ese modelo. A medida que la fábrica logra enviar autos, éstos se entregan a los clientes siguiendo el orden de prioridad de esa lista: por cada auto que llega, se busca el primer cliente de la lista que quiera ese modelo y que coincida en uno de sus dos colores.

a) Escriba un programa que permita primero cargar la lista de espera. Se cargan N clientes en orden de llegada (N no supera 200), y por cada cliente se ingresan 4 datos: su apellido y nombres (**nombre_completo**), el modelo de auto que quiere (**modelo**), y los dos posibles colores (**color_1** y **color_2**).

b) Luego, en una segunda etapa el programa debe permitir ingresar los autos que la fábrica envía, y por cada uno, determinar a qué cliente se le debe asignar. Si se recibe una combinación de modelo y color que ningún cliente pidió, mostrar solo un mensaje alusivo. Si se recibe una combinación que algún cliente haya pedido, se debe informar el nombre del cliente, y modificar la lista de espera de alguna forma para saber que ese cliente ya no está esperando y no volverlo a considerar. Esta etapa termina cuando se ingresa "fin" en lugar del modelo.

c) Mostrar la lista con los nombres de clientes a los que aún siguen en espera.

*ayuda: En el punto a, generar 4 listas (**nombres**, **modelos**, **colores_1**, **colores_2**) o una matriz de Nx4. En el punto b, buscar cada par modelo y color en esas listas, y cuando se encuentre cambiar el modelo a una cadena vacía u otro valor especial para indicar que ya no está en espera. En el punto c, mostrar los clientes que no tengan el valor especial en su modelo.*

 **Ejemplo de resolución:** https://youtu.be/EEEnTkR_Q4Jo?t=5048

Fundamentos de Programación 2021. 1er Parcial. 21/05/2021. Tema B

Ej 1 (35 pts) Un desarrollador quiere analizar la participación en foros relacionados a uno de sus programas. Hay 5 foros relacionados a distintos aspectos de ese programa.

a) Escriba un programa que permita cargar los datos de cada foro por cada mes del año. La carga de datos se debe realizar ingresando las cantidades, en orden por mes (primero las 5 cantidades de enero, luego las 5 de febrero, etc).

El programa luego debe informar:

b) El mes (**mes_mas_mensajes**) que más mensajes llegaron (sumando todos los foros) y cuántos mensajes fueron (**cant_mensajes**)

c) En cuántos y cuáles meses crecieron las cantidades de mensajes totales (sumando todos los foros). Es decir, meses en que la suma fue mayor que el mes anterior.

d) Por cada foro, el **promedio** de mensajes mensual.

 **Ejemplo de resolución:** <https://youtu.be/28sLt1Xkqec?t=142>

Ej 2 (35 pts) Un paseador de perros pasea 15 perros cada semana. Cada perro suele dar un paseo por día. El paseador quiere registrar los minutos que pasea con cada perro para asegurarse de que todos paseen lo suficiente de acuerdo a su tamaño. Se requiere un programa que:

a) Permita ingresar primero el tamaño de cada perro. El programa debe leer los tamaños en orden (primero para el perro 1, luego para el perro 2, etc). Por cada perro se lee una palabra que puede ser **grande**, **mediano** o **pequeño**.

b) Debe permitir ingresar luego los datos de los paseos de una semana. Por cada paseo que realiza con un perro se ingresan 2 datos: el número de perro (**perro**, de 1 a 10) y la cantidad de minutos (**minutos_semanales**) que duró. Serán en total 15*7 pares de datos.

c) El programa debe mostrar, por cada perro, los minutos totales caminados (**minutos_totales**), e informar con un mensaje alusivo si esa cantidad es suficiente para su tamaño. Para los perros chicos, se esperan un promedio de 20 min por día, así que la suma semanal debe ser de al menos 140 minutos. Para los medianos, 40 min por día, lo que da un total de 280 min por semana. Para los grandes, 1h or día, lo que da un total de 420 min por semana.

 **Ejemplo de resolución:** <https://youtu.be/28sLt1Xkqec?t=2332>

Ej 3 (30 pts) Un DT de un equipo de fútbol se cansó de tener que replantear la formación tantas veces antes de un partido por culpa de la bajas por covid, y decidió contratar un programador para hacer un software que la obtenga automáticamente. El programa debe funcionar de la siguiente manera:

a) Primero debe permitir al entrenador ingresar los datos de todos sus jugadores. Por cada uno de los **N** jugadores, el entrenador ingresa su nombre (**nombre_completo**) y su posición (**delantero**, **mediocampista**, **defensor**, **arquero**). El entrenador los cargará ordenados según sus preferencias: el primero en la lista será el que considera mejor jugador, el último el peor.

b) Luego, el programa debe permitir cargar las bajas por covid (una conjunto de nombres). El programa deberá modificar la lista de jugadores de alguna forma para luego saber que no están disponibles. Esta etapa finaliza cuando en lugar del nombre se ingresa "fin".

c) Finalmente, el programa debe mostrar los 3 mejores delanteros disponibles, los 4 mejores mediocampistas, los 3 mejores defensores, y el mejor arquero. Recuerde que los "mejores disponibles" serán los que aparezcan primeros en la lista, sin contar a los dados de baja. Si en alguna posición no hay suficientes jugadores disponibles, debe mostrar un mensaje alusivo.

*Ayuda o sugerencia: en el punto (a), generar una lista de **nombres** y otra de **posiciones**. En el (b) reemplazar las posiciones para los jugadores que se dan de baja una cadena en blanco o algún otro valor especial. En el punto (c), por cada posición, buscar los jugadores que no tengan ese valor especial en la posición, e ir contándolos y mostrándolos.*

 **Ejemplo de resolución:** <https://youtu.be/28sLt1Xkqec?t=3549>