

Apellido y Nombre: _____ Curso: _____ **T1****Ejercicio 1 (3 pts)**

Se lanzó la preventa del Primavera Sound 2023, la modalidad que se elige es lanzar una preventa de entradas, sin anunciar los artistas que se presentarán. Los early birds #1 (preventa de entradas), se lanzarán el 1 de mayo de 2023 a las 13:00 hs, pero los fanáticos pueden hacer “fila virtual” desde las 00:00 para asegurar su lugar en el famoso festival, una vez que se lance la venta virtual los usuarios pueden ir comprando las entradas de a tandas.

Con la finalidad de analizar si se debe optimizar el rendimiento de la plataforma, se disponen **dos secuencias de caracteres**. La primera secuencia contiene a los fanáticos que hacen la “fila virtual”, esta secuencia contiene la **hora** en la que se registró en la fila (hhmm), el **número en la fila** (6 caracteres), el **medio de pago** con el que piensa pagar (“T”- tarjeta, “C”-efectivo, “E”-transferencia) y **cuántas entradas** piensa comprar (como máximo puede comprar 5 entradas).

En estos sistemas de fila, al colocarse en ella se obtiene un identificador o número de fila, pero recién el usuario inicia sesión al llegar a su lugar en la fila y entrar a la sección de Compras.

En una segunda secuencia se encuentran las **compras reales realizadas**, esta secuencia contiene el **número en la fila** (6 caracteres), el **nombre de usuario** (cantidad indefinida, termina en “+”), y luego contiene los datos de para quién es cada entrada en la compra, incluyendo su DNI y nombre (separados con ‘.’). Una compra finaliza con un “?”.

Secuencia Fila_Virtual:

horafilenumerofilamediopagocantentradas#

Ejemplo:

0301123456T3#0345234567E4#

Secuencia Compras:

numerofilanombre_usuario+dninombre.dninombre.dninombre?

Ejemplo:

123456usuario+33254787Juan.27895614Melisa.36257489Pedro?

Si por algún motivo algún fanático deja su lugar en la fila, en la secuencia de **Compras** en nombre de usuario aparece un carácter “#”, el signo “+” y luego la marca “?”. Esto implica una correspondencia 1 a 1 entre las dos secuencias.

Escriba un algoritmo en pseudocódigo que resuelva las siguientes consignas:

- Generar una secuencia de salida con los nombres de usuario y cada DNI de las entradas compradas que fueron pagadas con Tarjeta. Separar cada campo con un + y finalizar con “#” al completar la compra de un usuario.
EJ: nombreusuario+dni+dni#nombreusuario+dni+dni#
- Se desea conocer cuántos usuarios compraron una cantidad de entradas distinta a la que declararon en la fila virtual.

Ejercicio 2 (2 pts)

Basados en el escenario del ejercicio 1, se tiene un archivo secuencial que contiene las informaciones de ventas finales de entradas para el festival, realizadas desde el 1 de mayo del 2023 hasta el 1 de julio de 2023.

VENTAS ordenado por provincia, ciudad, plataforma y fecha

provincia	ciudad	plataforma	fecha	entradas
-----------	--------	------------	-------	----------

Escribir un algoritmo en pseudocódigo que cumpla con las siguientes consignas:

- Generar un archivo de salida que contenga datos de las ciudades en las cuales la cantidad total de entradas vendidas supera las 1000, con el siguiente formato:

provincia	ciudad	entradas
-----------	--------	----------
- Informar el total de ventas del 1 de junio al 1 julio, discriminado por plataforma y ciudad.

Apellido y Nombre: _____ Curso: _____ **T2**

Ejercicio 1 (3 pts)

Se lanzó la preventa del Primavera Sound 2023, la modalidad que se elige es lanzar una preventa de entradas, sin anunciar los artistas que se presentarán. Los early birds #1 (preventa de entradas), se lanzarán el 1 de mayo de 2023 a las 13:00 hs, pero los fanáticos pueden hacer “fila virtual” desde las 00:00 para asegurar su lugar en el famoso festival, una vez que se lance la venta virtual los usuarios pueden ir comprando las entradas de a tandas.

Con la finalidad de analizar si se debe optimizar el rendimiento de la plataforma, se disponen **dos secuencias de caracteres**. La primera secuencia contiene a los fanáticos que hacen la “fila virtual”, esta secuencia contiene la **hora** en la que se registró en la fila (hhmm), el **número en la fila** (6 caracteres), el **medio de pago** con el que piensa pagar (“T”- tarjeta, “C”-efectivo, “E”-transferencia) y **cuántas entradas** piensa comprar (como máximo puede comprar 5 entradas).

En estos sistemas de fila, al colocarse en ella se obtiene un identificador o número de fila, pero recién el usuario inicia sesión al llegar a su lugar en la fila y entrar a la sección de Compras.

En una segunda secuencia se encuentran las **compras reales realizadas**, esta secuencia contiene el **número en la fila** (6 caracteres), el **nombre de usuario** (cantidad indefinida, termina en “+”), y luego contiene los datos de para quién es cada entrada en la compra, incluyendo su DNI y nombre (separados con ‘.’). Una compra finaliza con un “?”.

Secuencia Fila_Virtual:

horafilanumerofilamediopagocantentradas#

Ejemplo:

0301123456T3#0345234567E4#

Secuencia Compras:

numerofilanombre_usuario+dninombre.dninombre.dninombre?

Ejemplo:

123456usuario+33254787Juan.27895614Melisa.36257489Pedro?

Si por algún motivo algún fanático deja su lugar en la fila, en la secuencia de **Compras** en nombre de usuario aparece un carácter “#”, el signo “+” y luego la marca “?”. Esto implica una correspondencia 1 a 1 entre las dos secuencias.

Escriba un algoritmo en pseudocódigo que resuelva las siguientes consignas:

- Generar una secuencia que contenga el número de fila y DNI para cada entrada comprada por esa persona, solo de aquellas compras que entraron en la cola antes de las 10:00 am. con el siguiente formato.
EJ: numerofila+dni+dni#numerofila+dni+dni#
- Se desea conocer la cantidad de usuarios que estaban en la fila, pero no compraron entradas (dejaron la fila).

Ejercicio 2 (2 pts)

Basados en el escenario del ejercicio 1, se tiene un archivo secuencial que contiene las informaciones de ventas finales de entradas para el festival, realizadas desde el 1 de mayo del 2023 hasta el 1 de julio de 2023.

VENTAS ordenado por provincia, ciudad, plataforma y fecha.

| provincia | ciudad | plataforma | fecha | entradas |

Escribir un algoritmo en pseudocódigo que cumpla con las siguientes consignas:

- Generar un archivo de salida con todas las entradas vendidas en el mes de junio con el siguiente formato:
| provincia | entradas |
- Informar las ciudades (y la provincia a la que pertenecen) que tengan una venta superior a las 7500 entradas.

Apellido y Nombre: _____ Curso: _____ **T1**

Ejercicio N° 1 (3 pts)

Un importante supermercado de la provincia del Chaco posee la información del stock de todos sus artículos en una secuencia de datos, con la siguiente estructura:

CodArtCodRubroStockNombreArticulo&CodArtCodRubroStockNombreArticulo&CodArtCodRubroStockNombreArticulo&FDS

Donde:

- CodArt: (5 caracteres) código del artículo.
- CodRubro: (1 carácter), se refiere al rubro del artículo, las opciones son:
"L": Limpieza, "F": Fiambrería, "C": Carnicería, "B": Bazar, "H": Higiene
- Stock: 3 caracteres, cantidad de artículos en stock.
- NombreArticulo: es el nombre del artículo y finaliza con un "&"

Ejemplo: 12345L789Detergente Magistral&23456F078Jamon Iberico& [...] &FDS

Además, se posee una secuencia de caracteres con todas las ventas realizadas para los artículos (el fin de las ventas de cada artículo se indica con el carácter "#").

DiaMesFPFEUVDiaMesFPFEUVDiaMesFPFEUV#DiaMesFPFEUVDiaMesFPFEUVDiaMesFPFEUV# [...] #

DiaMesFPFEUVDiaMesFPFEUVDiaMesFPFEUV # [...] FDS

Donde:

- Dia: (2 caracteres) corresponde al día de la venta.
- Mes: (2 caracteres) corresponde al mes de la venta.
- FP: (1 carácter) indica forma de pago: "T": Tarjeta de crédito – "C": Contado.
- FE: (1 carácter) indica forma de envío: "S": Entregado en sucursal - "D": Envío a domicilio.
- UV: (2 caracteres) Unidades Vendidas.

Existe una correspondencia uno a uno entre las 2 secuencias, de la siguiente forma: el primer grupo de ventas corresponde al primer artículo, el siguiente al segundo, y así sucesivamente.

Se lo ha contratado a usted para desarrollar una solución en pseudocódigo que permita:

- Generar una nueva secuencia de salida con los nombres de todos los artículos que han quedado sin stock (stock = 0). Para poder determinar el stock de un producto sólo se deberán descontar las unidades cuya forma de envío haya sido "Entregado en sucursal".
- Generar un informe de las ventas realizadas para un determinado mes que ingresa un usuario, con la siguiente estructura:
Nombre del Artículo | Cant. unid entregadas en suc | Cant. unid enviadas a domicilio

Ejercicio 2 (2 pts)

La misma cadena de supermercados, además cuenta con un archivo secuencial con el stock de todos sus artículos con el siguiente formato:

Stock, ordenado por Código sucursal, Rubro y Código Artículo.

Cod Suc N(2)	Rubro AN(20)	Cod Artículo N(5)	FechaUltRep	Stock de seguridad	Stock actual
--------------	--------------	-------------------	-------------	--------------------	--------------

FechaUltRep: fecha última reposición

El stock de seguridad es el nivel mínimo de existencias que se debe mantener en almacén.

Se le pide escribir un algoritmo en pseudocódigo que permita:

- Generar un informe que muestre todos los artículos (sucursal, rubro y código de artículo) cuyo stock actual esté por debajo del stock de seguridad, indicando cantidad total por sucursal, por rubro y total general.
- Generar un archivo de salida que contenga la cantidad total de artículos que requieren reposición (stock actual menor a stock de seguridad) por Rubro, por cada sucursal. Debe contener: código sucursal, rubro y cantidad de artículos.

Apellido y Nombre: _____ Curso: _____ **T2**

Ejercicio N° 1 (3 pts)

Un importante supermercado de la provincia del Chaco posee la información del stock de todos sus artículos en una secuencia de datos, con la siguiente estructura:

CodArtCodRubroStockNombreArticulo&CodArtCodRubroStockNombreArticulo&**CodArtCodRubroStockNombreArticulo&**FDS

Donde:

- CodArt: (5 caracteres) código del artículo.
- CodRubro: (1 carácter), se refiere al rubro del artículo, las opciones son:
“L”: Limpieza, “F”: Fiambrería, “C”: Carnicería, “B”: Bazar, “H”: Higiene
- Stock: 3 caracteres, cantidad de artículos en stock.
- NombreArticulo: es el nombre del artículo y finaliza con un “&”.

Ejemplo: 12345L789Detergente Magistral&23456F078Jamon Iberico& [...] &FDS

Además, se posee una secuencia de caracteres con todas las ventas realizadas para los artículos (el fin de las ventas de cada artículo se indica con el carácter “#”).

DiaMesFPFEUVDiaMesFPFEUV**DiaMesFPFEUV#**DiaMesFPFEUV**DiaMesFPFEUV#**DiaMesFPFEUV**DiaMesFPFEUV#** [...] #

DiaMesFPFEUV**DiaMesFPFEUV#** [...] FDS

Donde:

- Dia: (2 caracteres) corresponde al día de la venta.
- Mes: (2 caracteres) corresponde al mes de la venta.
- FP: (1 carácter) indica forma de pago: “T”: Tarjeta de crédito – “C”: Contado.
- FE: (1 carácter) indica forma de envío: “S”: Entregado en sucursal - “D”: Envío a domicilio.
- UV: (2 caracteres) Unidades Vendidas.

Existe una correspondencia uno a uno entre las 2 secuencias, de la siguiente forma: el primer grupo de ventas corresponde al primer artículo, el siguiente al segundo, y así sucesivamente.

Se lo ha contratado a usted para desarrollar una solución en pseudocódigo que permita:

- Generar una nueva secuencia de salida con los nombres de todos los artículos que tengan mayor cantidad de ventas con tarjeta de crédito que al contado.
- Generar un informe de las ventas realizadas para una determinada forma de envío que ingresa el usuario, con la siguiente estructura:
Nombre del Artículo | Cant. unidades vendidas con tarjeta | Cant. unid vendidas al contado.

Ejercicio 2 (2 pts)

La misma cadena de supermercados, además cuenta con un archivo secuencial con el stock de todos sus artículos con el siguiente formato:

Stock, ordenado por Código sucursal, Rubro y Código Artículo

Cod Suc N(2)	Rubro AN(20)	Cod Articulo N(5)	FechaUltRep	Stock de seguridad	Stock actual
--------------	--------------	-------------------	-------------	--------------------	--------------

FechaUltRep: fecha última reposición

El stock de seguridad es el nivel mínimo de existencias que se debe mantener en almacén.

Se le pide escribir un algoritmo en pseudocódigo que permita:

- Generar un informe de totales por sucursal, por rubro y total general de cantidad de artículos cuya fecha de última reposición sea anterior a una fecha ingresada por el usuario.
- Generar un archivo de salida que contenga todos los artículos del rubro “Bazar”, cuya fecha de última reposición sea anterior a la fecha ingresada por el usuario. Debe contener sucursal y código de artículo.