

**NOMBRE:**

Lea la prueba completamente **DOS** veces antes de hacer cualquier pregunta

Suba el archivo .py de cada problema a las tareas dispuestas en Campus Virtual. Las entregas pasadas la hora no serán evaluadas y tendrán la nota mínima. Considere que se puede demorar al subir los archivos, así que use el tiempo de entrega para realmente entregar los archivos, y no para continuar programando. Si el código no contiene el encabezado indicado (su nombre como comentario dentro de las primeras 5 líneas de cada archivo), se descontará un 35% del puntaje obtenido en cada uno de los ejercicios donde no cumplió la instrucción. Una prueba respondida correctamente en un 60% corresponde a una nota 4,0. En su espacio personal no debe haber nada más que hojas de papel y lápiz. El resto de sus implementos debe guardarlos dentro de su mochila/bolso y ésta debe posicionarse al frente debajo de la pizarra. Si leyó hasta aquí, felicidades. Ahora para saber que lo hizo escriba “voy a aprobar el curso” al final de esta página.

**SUBA SU AVANCE AL CAMPUS VIRTUAL PERIODICAMENTE EN CASO DE FALLO DEL EQUIPO.**

## Problema A. Comprando PCs (70%)

Una tienda recientemente inaugurada, que vende partes y piezas de computadores está comenzando sus operaciones, pero no tiene algún sistema informático que les permita controlar las ventas de sus productos. Ellos tienen una filosofía diferente a las demás empresas del rubro, se enfocan en 1 solo producto para tres gamas de precios, gama alta, media y baja. Los productos que venden son, Procesadores, GPUs, Placas Madre, SSD, y RAM.

Ellos te entregarán algunos datos para que puedas crear un programa que les ayude a saber cuántos ingresos generan y si las ordenes de compra que reciben son posibles de cumplir.

El primer archivo que te entregarán será **stock.txt**, que tiene la información de los productos que hay disponibles. La estructura es la siguiente:

**parte,cantidad,gama,precio**

donde

**parte:** corresponde al nombre del componente. Cada componente puede aparecer más de una vez, ya que puede ser una reposición de inventario.

**cantidad:** es el número de partes que se está ingresando.

**gama:** puede ser alta, media o baja.

**precio:** corresponde al valor unitario en dólares de la parte. Si anteriormente ya existía el mismo producto, el precio considerado para la venta será el último que aparezca para ese componente.

Por otra parte, recibirás un archivo llamado **orden\_compra.txt** que tiene la siguiente estructura.

**oc,parte,cantidad,gama**

donde **oc** será el código identificador de la orden de compra, **parte** será el producto solicitado, junto a la **cantidad** y la **gama**.

Teniendo esto se te pedirá realizar lo siguiente:

- 1) Indicar cuales ordenes de compra no pueden ser realizadas. Una orden no puede ser procesada cuando no hay stock suficiente para cubrir la cantidad solicitada o cuando la cantidad requerida supera las 15 unidades.
- 2) Ingresos según gama de productos.
- 3) Indicar qué productos necesitan reposición. Un producto necesita reposición si la cantidad en stock es inferior a 20.

4) Precio promedio por producto. Considere para este cálculo el precio actual de los productos.

Ejemplo de ejecución

| stock.txt   | Salida por pantalla   |
|---|---|
| Procesador,18,baja,85<br>GPU,51,alta,1100<br>Placa Madre,53,baja,40<br>Placa Madre,34,baja,35<br>SSD,23,media,60<br>GPU,32,media,500<br>GPU,35,baja,250<br>GPU,48,media,529<br>GPU,34,media,535<br>Placa Madre,41,alta,320<br>Placa Madre,50,alta,320<br>Procesador,37,alta,650 | 1) Ordenes no válidas<br>orden 981151825-H no se puede procesar<br>orden 353202736-O no se puede procesar<br>orden 412491154-L no se puede procesar<br>orden 312923329-A no se puede procesar<br>orden 163095270-G no se puede procesar<br>orden 163095270-F no se puede procesar<br>2) Ingresos<br>- alta: 57410<br>- media: 32420<br>- baja: 14586<br>3) Reposición de productos<br>- reponer GPU de gama baja<br>- reponer Placa Madre de gama media<br>- reponer RAM de gama baja<br>4) Precio promedio por producto:<br>- Procesador: 369.0<br>- GPU: 631.67<br>- Placa Madre: 201.67<br>- SSD: 115.0<br>- RAM: 80.0 |
| orden_compra.txt  |   |
| 638130829-X,RAM,1,media<br>343392021-4,RAM,1,alta<br>981151825-H,RAM,100,media<br>820249676-4,Placa Madre,2,alta<br>307988714-X,Procesador,2,baja<br>802071901-6,Procesador,2,media<br>942614910-4,GPU,3,alta<br>812216900-7,GPU,2,alta<br>438814172-0,RAM,2,alta<br>...        |   |

Problema B. Usando subprogramas (30%)

Se te entregará un archivo que se llama **autos.txt** y tiene el siguiente formato:

**vin,marca,modelo,año,color,kilometraje**

En base a este archivo tendrás que realizar los siguientes subprogramas.

- 1) **Procedimiento** que reciba al menos una **matriz** (del kilometraje acumulado según marca y año de los vehículos), un **año** y que imprima el kilometraje promedio por año desde el año pasado como parámetro. Los años deben aparecer en forma ascendente.
- 2) **Función** que reciba al menos una **lista** del **VIN** de los vehículos junto con una del **kilometraje** asociado y un valor **N** y que con eso retorne los N VIN con mayor kilometraje recorrido.
- 3) **Procedimiento** que reciba al menos una **matriz de cantidad de modelos por color** y que retorne los modelos que tienen más de 2 auto con el mismo color.

Consideraciones

- No habrá más de 100 marcas ni años.
- No habrá más de 600 modelos de vehículo diferente.

Ejemplo de ejecución

|  |
|--|
| <b>Estacionados.txt (archivo completo en el Campus Virtual)</b>  |
| 1G4HP54K21U008453,Pontiac,GTO,2005,Green,12855<br>WBA3D5C57EK128170,Chevrolet,Avalanche 1500,2004,Mauv,224064<br>3C4PDDGG1FT116171,Ford,Taurus,2011,Red,172814<br>WAUDF78E28A805039,Audi,S4,2004,Puce,210540<br>1FTMF1CW9AK295642,Jeep,Grand Cherokee,2007,Violet,187607<br>1G6AJ1R32F0943863,Mitsubishi,Pajero,1987,Mauv,200809<br>WBAUT9C53AA095411,Chevrolet,Uplander,2006,Khaki,215801<br>WBAVB13596P856584,Ford,Ranger,2004,Puce,225446<br>2G4WF521541323378,Hyundai,Tiburon,2005,Pink,204673<br>WAUAGAFD0EN410868,Lamborghini,Countach,1990,Fuscia,135612<br>2G4GT5GK8C9886939,Volvo,S80,2012,Pink,10758<br>1FTMF1E83AF132203,Toyota,RAV4,2005,Purple,217344<br>TRUTC28N331133002,Pontiac,Grand Prix,1990,Orange,52232<br>WBALZ5C5XCD801780,Ford,Thunderbird,1997,Purple,47943<br>JM3TB2MA9A0406017,BMW,X6 M,2011,Green,243968<br>WBALM5C50BE443452,Mercedes-Benz,M-Class,2011,Mauv,185709<br>SCBZB25E31C978867,Volkswagen,Passat,1986,Maroon,176125 |
| <b>Salida</b>  |
| 1) Kilometraje promedio desde 1990<br>1990 174770<br>1991 366968<br>1992 215926<br>1993 233321<br>1994 216985<br>1995 216558<br>1996 243665<br>1997 254379<br>1998 181730<br>1999 222187<br>2000 218985<br>2001 207374<br>2002 258674<br>2003 269567<br>2004 321910<br>2005 226608<br>2006 262119<br>2007 248646<br>2008 246513<br>2009 194500<br>2010 163633<br>2011 255269<br>2012 228002<br>2013 161341<br>2) Los 3 kilometrajes mayores<br>['5J8TB4H33GL428880', '19XFB4F2XDE711237', 'WBABW33444P874336']<br>3) Colores repetidos<br>modelo: DeVille color: Puce<br>modelo: Cabriolet color: Fuscia<br>modelo: GTI color: Fuscia  |

**Prueba 2 – Programación – Invierno 2022**

**Duración:** 4 horas

19 de agosto 2022

**Con el fin de que al revisar su prueba se entienda mejor lo que realizó, utilice nombres de variables que tengan relación con el propósito que tienen. Si va a crear un contador, prefiera nombres como “contador” o “cont” y evite nombres como “c”. Así su profesor no tendrá problemas en interpretar su código.**

**Compromiso de honestidad**

Me comprometo a ser honesto al realizar esta evaluación, y a que entregaré el resultado de mi trabajo personal.

---

Nombre, RUT y Firma

(Debe entregar este documento firmado antes de retirarse)