

Escenario: Laboratorio Control de Calidad

Una empresa siderurgia, diariamente se dedica a producir alambρόn con una especificación de porcentaje en peso de Carbono (%C) y Magnesio (%Mg) en particular, dicha empresa necesita que se le elabore un programa que automatice el proceso de aprobación o no de cada lote de producción realizado en el día. Para ello diariamente el laboratorio coloca la información en el archivo de datos “**Lotes.txt**”, en el cual en la primera línea tenemos:

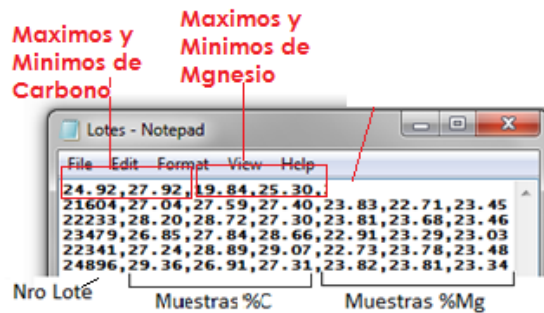
Porcentaje de C mínimo, Porcentaje de C máximo, Porcentaje de Mg mínimo, Porcentaje de Mg máximo

En las líneas siguientes:

Numero del lote evaluado y los resultados de las tres muestras enviadas para determinar el porcentaje en peso de Carbono y Magnesio respectivamente.

Ver figura mostrada.

La política de la empresa para aprobar o no un lote es que el promedio de las tres muestra evaluadas debe ser mayor que el porcentaje mínimo y menor al porcentaje máximo establecido para cada elemento (C y Mg).



Enunciado:

Elabore un programa en Python, que lea el archivo de datos “Lotes.txt” y determine e imprima:

- Por cada lote:
 1. Número del Lote, Promedio del %peso de C, Promedio del %peso de Mg y su estatus (Aprobado o Rechazado)
- Adicionalmente, para todos los lotes:
 2. Porcentaje de lotes rechazados en el día.
 3. Monto perdido, sabiendo que por cada lote rechazado se pierde 2500.00 \$
- La salida por pantalla quedara formateada tal como se muestra a continuación:

```
Nro Lote    Prom %C    Prom %Mg    Estatus
21604       27.34      23.33      Aprobado
22233       28.07      23.65      Rechazado
23479       27.78      23.08      Aprobado
22341       28.40      23.33      Rechazado
24896       27.86      23.66      Aprobado
*****

Calculos finales y salida resultados
Porcentaje de lotes rechazados: 40.00%
Monto de las perdidas lotes rechazados 5000.0 $
```

Restricciones:

1. Todos los elementos de los vectores y matrices tiene el mismo tipo de datos.
2. El tamaño del arreglo es fijo, no se puede alterar.

Requerimientos mínimos:

1. Se debe definir las siguientes funciones:

- Que lea el contenido del archivo "Lotes.txt", tendrá como parámetros: una referencia al archivo "Lotes.txt", un vector de tipo **float** donde se almacenaran los máximos y mínimos de los porcentajes de Carbono (**C**) y Magnesio (**Mg**), un vector de tipo **String** donde se almacenaran los números de los lotes y una matriz de tipo **float** donde se almacenaran los porcentaje de las muestras de **C** y **Mg** tomadas de cada lote, colocando todo el contenido del archivo en el arreglo correspondiente.
- Que calcule el promedio de los porcentajes de **C** y **Mg** de cada lote, tendrá como parámetro la matriz con los porcentaje de C Y Mg tomados de cada lote, estos promedios serán almacenados en una matriz de tipo float que será retornada por la función.
- Que calcule el estatus de cada lote, tendrá como parámetro la matriz con los promedios de C y Mg de cada lote, y el vector donde se almaceno los máximos y mínimos de **C** y **Mg**, la función retornara un vector donde se guardara por cada lote si este fue "Aprobado" o "Rechazado".
- Que muestre por consola el número del lote, los promedios de **C** y **Mg** y el estatus de cada lote, tendrá como parámetros: el vector con los lotes, la matriz con los promedios, y el vector con el estatus de cada lote. Adicionalmente la función debe calcular y mostrar por consola el porcentaje de lotes rechazados así como monto total en dólares que representa estos lotes, sabiendo que cada lote rechazado tiene un coste de **2500 \$**. La salida debe quedar como se muestra en la figura:

```
Analisis de las Muestras
Lote      Prom %C      Prom %Mg      Estatus
21604     27.34      23.33      Aprobado
22233     28.07      23.65      Rechazado
23479     27.78      23.08      Aprobado
22341     28.40      23.33      Rechazado
24896     27.86      23.66      Aprobado

Porcentaje de lotes rechazados: 40.00
Monto de las perdidas lotes rechazados: 5000 $
```

- No se permiten definir funciones adicionales a las indicadas arriba, se deben usar los operadores lógicos cuando se requiera.
- El código principal debe definir:**
 - Una referencia al archivo a usar.
 - Determinar la cantidad de registros en el archivo "Lotes.txt"
 - Retornar el puntero del archivo al primer registro (usar la función **refArchivo.seek(0)**).
 - Definir los vectores para guardar las tolerancias, los números de los lotes y la matriz que guarda las muestras, usando el procedimiento de listas por comprensión.
 - E invocar los métodos que lee el archivo, que calcula los promedios, que define el estatus y que muestra los resultados por consola.

Código Respuesta:

```
1  # Definicion de funciones
2  # Funcion lee el archivo de datos
3  def leoLab(refArch,tol,lote,muestras):
4      linea = refArch.readline() # lee el 1er registro del archivo
5      campos = linea.split(",") # Lo fragmenta en cada uno de sus campos
6      for i in range(4): tol[i]= float(campos[i]) # Asigna la informacion al vector tol
7      f = 0
8      for linea in refArch: # Ciclo para leer el resto de los registros del archivo
9          campos = linea.split(",")
10         lote[f] = campos[0] #Asignamos el nro de lote al vector lote
11         for c in range(6): # Ciclo recorre los % de C y Mg
12             muestras[f][c] = float(campos[c+1]) #Asigna los % leidos en matriz muestras
13         f+=1
```

```
14 # Funcion para calcular los promedios
15 def promediosC_Mg(muestras):
16     # Definimos la matriz prom
17     prom = [[None]*2 for i in range(len(muestras))]
18     cp = 0
19     #Ciclo para recorrer las filas matriz muestras
20     for f in range(len(muestras)):
21         sumaC = 0
22         sumaMg = 0
23         #Ciclo para recorrer las columnas matriz muestras
24         for c in range(len(muestras[0])):
25             if c < 3:
26                 sumaC += float(muestras[f][c]) #Acumula los %C
27             else:
28                 sumaMg += float(muestras[f][c]) #Acumula los %Mg
29             #Calculamos los promedio de los %C y %Mg
30             prom[f][cp] = sumaC/3
31             prom[f][cp+1] = sumaMg/3
32     return prom
33
34 # Funcion para determinar el estatus de cada lote
35 def estatusMuestras(tol,prom):
36     #Definimos el vector estatus
37     est = [None for i in range(len(prom))]
38     c = 0 #variable para fijar la columna de la matriz prom
39     # Ciclo recorre las filas matriz prom
40     for f in range(len(prom)):
41         #Evaluamos los promedio de los % de C y Mg de acuerdo a las tolerancias
42         if (prom[f][c]>tol[0] and prom[f][c]<tol[1]) and (prom[f][c+1]>tol[2] and prom[f][c+1]<tol[3]):
43             est[f] = "Aprobado"
44         else:
45             est[f] = "Rechazado"
46     return est
47
48 # Funcion para mostrar los resultados finales
49 def mostrar(lotes,prom, est):
50     contR = 0
51     por = 0.0
52     print("\tAnálisis de las Muestras")
53     print("Lote      Prom %C      Prom %Mg      Estatus")
54     for f in range(len(prom)):
55         print("{0:8}".format(lotes[f]), end=" ")
56         for c in range(len(prom[0])):
57             print("{0:5.2f} ".format(prom[f][c]),end=" ")
58         print("{0:10}".format(est[f]))
59         if est[f] == "Rechazado":
60             contR += 1
61     if contR > 0:
62         por = contR/len(lotes) * 100
63         print("\nPorcentaje de lotes rechazados: {0:5.2f}".format(por))
64     else:
65         print("No se rechaza ningun lote")
66     print("Monto de las perdidas lotes rechazados: ", (contR*2500), " $")
67
```

```
68 #Codigo Principal principal
69 archivo = open("Lotes.txt") #Abrimos el archivo asignando la referencia a la variable archivo
70 # Determinamos la cantidad de registros
71 cantReg = len(archivo.readlines())
72 # La intruccion anterior coloca el puntero en el ultimo registro
73 # Con funcion seek colocamos el puntero en el 1er registro
74 archivo.seek(0)
75 # Definimos los vectores lotes y tole, y , la matriz muestras
76 lotes = [None for i in range(cantReg-1)]
77 tole = [None for i in range(4)]
78 muestras = [[None]*6 for i in range(cantReg-1)]
79 #Invocamos las funciones previamente definidas
80 leoLab(archivo,tole,lotes,muestras)
81 promedios = promediosC_Mg(muestras)
82 estatus = estatusMuestras(tole,promedios)
83 mostrar(lotes,promedios,estatus)
```