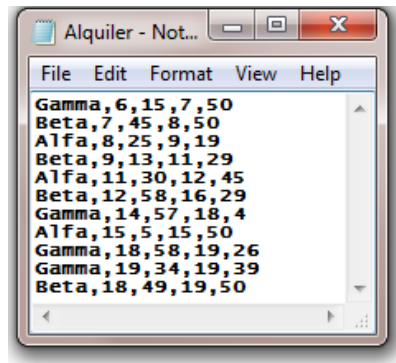


Escenario: Alquiler Computadoras en CyberCafe

Un centro de navegación de la ciudad, dispone de 3 (tres) máquinas para el alquiler por horas de uso, este centro registra en el archivo de datos **"Alquiler.Txt"**, por cada alquiler realizado en él día (ver figura):

**El nombre de la máquina, la hora de inicio y finalización del alquiler
(En horas y minutos).**

El Centro de Navegación genera el archivo a partir de las 08:00 y hasta las 18:00 de ese día, posteriormente dicho archivo se emplea para cuadrar la caja al cierre de actividades. La tarifa de cobro por alquiler es la siguiente: por cada hora cumplida 3 \$ y por cada fracción de 15 minutos 0.75 \$.



Enunciado:

Los dueños del centro lo contratan a usted para que elabore un programa en Python, que dado el archivo de datos **"Alquiler.Txt"**, determine e imprima:

Para cada alquiler:

1. Nombre de la maquina, tiempo de uso (expresado en Horas y Minutos), monto cobrado (Bs.).

Consideraciones:

- ❖ La hora de inicio y finalización se encuentra en formato de hora militar (Horas: Minutos).
- ❖ En el archivo sólo se registran los alquileres de un solo día de trabajo.
- ❖ La conversión de tiempo son: 1 hora = 60 Min.
- ❖ No se conoce cuantas veces fue alquilada cada máquina, así como el número de alquileres realizados en el día.
- ❖ La salida del programa debe quedar tal como lo muestra la figura (formateada):

Maquina	Tiempo Alquiler	Monto a Pagar
Gamma	01:35	05.25
Beta	01:05	03.75
Alfa	00:54	03.00
Beta	02:16	07.50
Alfa	01:15	03.75
Beta	03:31	11.25
Gamma	03:07	09.75
Alfa	00:45	02.25
Gamma	00:28	01.50
Gamma	00:05	00.75
Beta	01:01	03.75

- ❖ Modifique el programa de tal manera que determine e imprima la primera máquina cuyo al alquiler fue menor o igual a los 30 minutos. Respuesta: **Maquina 1 con 7 minutos.**

Requerimientos mínimos:

- A. Se deben definir las siguientes funciones:
 1. Que determine la cantidad de registros en el archivo **"Alquiler.txt"**, tendrá como parámetro el nombre del archivo.
 2. Que lea el contenido del archivo **"Alquiler.txt"**, recibirá como parámetros el nombre del archivo y una matriz de tipo entero donde almacenara los tiempos de inicio y terminó del alquiler, la función retornara un vector de tipo **String** con los nombres de las maquinas para cada uno de los alquileres.
 3. Que calcule la duración del alquiler, tendrá como parámetro la matriz de tipo entero con los tiempos de inicio y termino del alquiler, la función retornara dos valores la hora y los minutos que duro el alquiler.

4. Que calcule el monto del alquiler, tendrá como parámetro un vector con la hora y los minutos que duro el alquiler.
 5. Que retorne una matriz de enteros con los tiempos de duración del alquiler en horas y minutos, tendrá como parámetro la matriz de enteros que guarda los tiempos de inicio y termino del alquiler. Esta función invocara la función 3.
 6. Que retorne un vector con los montos de los alquileres, tendrá como parámetro la matriz con los tiempos de duración de cada alquiler en horas y minutos, aquí se invocara la función 4.
 7. Que muestre los resultados recibirá como parámetros el vector con los nombres de las maquinas, la matriz con la duración del alquiler y el vector con los montos de cada alquiler. La salida debe quedar formateada tal como se muestra en la figura de arriba.
- B. En el código principal:
- I. Se asignara el nombre del archivo.
 - II. Se invocara la función (1) que calcule la cantidad de registros que hay en el archivo.
 - III. Se creara la matriz que guardara los tiempos de inicio y de término del alquiler.
 - IV. Se invocara el función (5) para determinar la duración del alquiler, función
 - V. Se invocara la función (6) que calcule los montos del alquiler.
 - VI. Se invocara la función (7) que muestre los resultados finales

Respuesta:

```
# definicion de funciones
# Funcion determina la cantida de registros que hay en el archivo
def cantReg(Archivo):
    refArchivo = open(Archivo,"r")
    cant = len(refArchivo.readlines())
    refArchivo.close()
    return cant

# Funcion que lee el contenido del archivo
def leoAlquiler(Archivo, ti):
    refArchivo = open(Archivo, "r")
    fila = 0
    f = len(ti)
    c = len(ti[0])
    maq = [None for i in range(f)] # Creamos el vector
    for linea in refArchivo:
        campos = linea.split(",")
        maq[fila] = campos[0]
        for col in range(c):
            ti[fila][col] = int(campos[col + 1])
        fila += 1
    refArchivo.close()
    return maq

# Funcion que calcula la duracion del alquiler
def duracion(t):
    dura = (t[2]-t[0])*60 +(t[3]-t[1])
    hr = dura // 60
    mi = dura % 60
    return hr, mi
```

```
# Funcion que calcula el monto del alquiler
def montoAlg(t):
    monto = t[0]*3+ 0.75*(t[1]//15)
    if t[1]%15 > 0:
        monto += 0.75
    return monto

# Funcion genera una matriz con los tiempos de duracion alquiler
def vectorDura(ti):
    f = len(ti)
    c = len(ti[0])
    tidur = [[None] * c for i in range(f)] # Creamos la matriz
    for fila in range(f):
        tidur[fila][0],tidur[fila][1] = duracion(ti[fila])
    return tidur

# Funcion que genera un vector con los montos
def vecMontos(td):
    montos = [None for i in range(len(td))] # Creamos el vector
    for fila in range(len(td)):
        montos[fila]=montoAlg(td[fila])
    return montos

# Funcion mostrar resultados
def mostrar(maq,td,mo):
    for fila in range(len(td)):
        print("{0:7}          {1:02d}:{2:02d}          {3:05.2f}".
              format(maq[fila],td[fila][0],td[fila][1],mo[fila]))

# Programa Principal
# Abrir el archivo
archivo = "Alquiler.txt"
# determinamos la cantidad de registros en el archivo
reg = cantReg(archivo)
c = 4
tiempos = [[None] * c for i in range(reg)] #Creamos la matriz
maquinas = leoAlquiler(archivo,tiempos)
duracionAlquiler = vectorDura(tiempos)
montosAlquiler = vecMontos(duracionAlquiler)
print("Maquina      Tiempo Alquiler      Monto a Pagar")
mostrar(maquinas,duracionAlquiler,montosAlquiler)
```