

Nombre:

Cantidad de horas de estudio personal:
Esta semana
Semana pasada
Semana ante-pasada

Problema 1. Ruteo de código (20%)

Rutee el código de la derecha, e indique el valor que van tomando todas las variables. Escriba los valores usando una tabla como la siguiente:

Table with 4 columns: Nombre variable1, Nombre variable1, Nombre variable1, ...

En el encabezado de la tabla escriba el nombre de la variable, y hacia abajo escriba los valores que adquiere la variable a medida que van cambiando de valor. Escriba un valor por casillero. Use la siguiente página para escribir su tabla con valores.

```
lista = []

for i in range(2, 10, 3):
    lista.append(i)

for i in range(1, 10, 3):
    lista.append(i)

q = len(lista)

for a in range(q):
    for b in range(a, q):
        if lista[a] < lista[b]:
            t = lista[a]
            lista[a] = lista[b]
            lista[b] = t

m = -1
i = 0
su = 0

while i < q:
    su = su + m * lista[i]
    m = m + 1
    i = i + 1
    if m == 2:
        m = -1

print(su)
```

Problema2. Sistema de ecuaciones (40%)

En una ciudad se está realizando un estudio de impacto para tratar de determinar el efecto que producirá la construcción de un estadio cerca de la plaza.

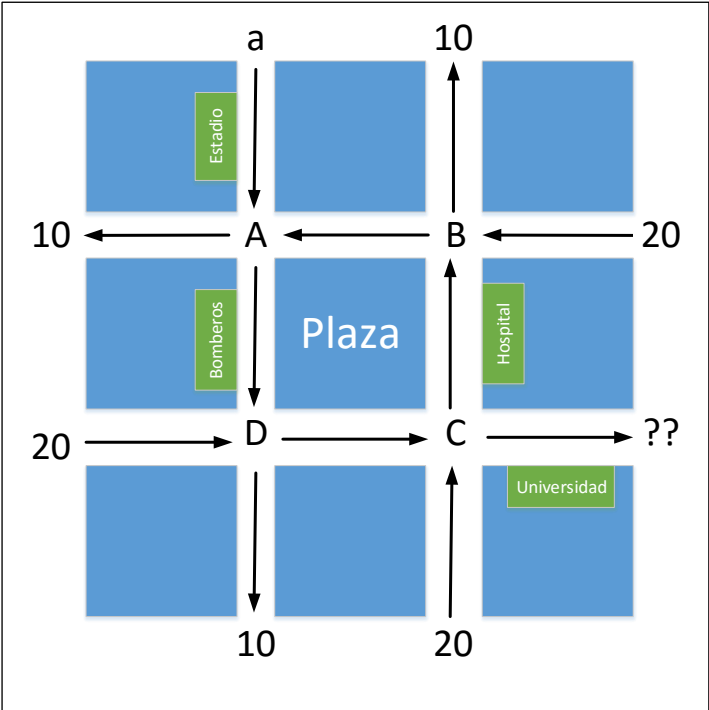
Durante un día normal, el tráfico en las calles cercanas a la plaza es prácticamente constante, como se indica en la figura. Los números indican la cantidad de vehículos en la calle donde está cada flecha, y cada flecha indica el sentido de la calle.

Cree un programa que a partir de la cantidad de vehículos que pasarán frente al estadio (valor a en la figura), calcule la cantidad de vehículos que pasarán frente a la Universidad (valor ??) y la cantidad de vehículos que pasarán frente a los Bomberos. El programa debe mostrar dichos valores con a variando entre 0 y 10.

- Empíricamente se ha calculado que el tráfico en la calle del Hospital es siempre el doble que el tráfico en la calle donde estará el estadio.
- En cada intersección (A, B, C y D) la suma del tráfico que entra debe ser igual a la suma del tráfico que sale.

Por ejemplo:

Table with 3 columns: a, Bomb, Salida



Problema3. Matrices (40%)

Un grupo de amigos se junta todos los sábados a jugar. El juego preferido de ellos es uno de carreras, en donde todos compiten en diferentes pistas y el ganador se determina de acuerdo al tiempo en completar el circuito.

- Construya un programa que realice los siguientes pasos:
- 1. Pregunte el nombre de todos los jugadores. Use FIN como terminador de ciclo. (8%)
 - 2. Después de haber preguntado todos los nombres, que pregunte el nombre de todas las pistas en las que se competirá. Use FIN como terminador de ciclo. (8%)
 - 3. Una vez ingresados los nombres y las pistas, que pregunte el tiempo de cada jugador en cada circuito. El programa debe indicar el nombre de la pista, y después preguntar el tiempo de cada jugador. Posteriormente, indicar el nombre de la siguiente pista y los tiempos de los jugadores. (17%)
 - 4. Una vez ingresados todos los datos, el programa debe escribir por pantalla lo siguiente:
 - a. Por cada pista, el nombre y tiempo del jugador más rápido (25%)
 - b. El nombre del jugador con promedio de tiempo más bajo (considerando todas las pistas donde corrió) (25%)
 - c. El nombre, pista y tiempo del tiempo más alto registrado (considerando todos los jugadores y todas las pistas) (17%)

- Observaciones:
- 1. No todos los jugadores compiten en todas las pistas. En ese caso, se ingresa un cero.
 - 2. La cantidad de jugadores y pistas es variable.
 - 3. Considere que siempre puede existir más de un elemento que cumpla el criterio de búsqueda (vea el ejemplo).
 - 4. En la siguiente página dibuje la estructura de datos que utilizará para almacenar los datos y resolver el problema.

```
Ingrese nombre: JugadorA
Ingrese nombre: JugadorB
Ingrese nombre: JugadorC
Ingrese nombre: FIN
Ingrese pista: Pista1
Ingrese pista: Pista2
Ingrese pista: Pista3
Ingrese pista: FIN

Datos pista Pista1
    Ingrese tiempo de JugadorA: 12
    Ingrese tiempo de JugadorB: 0
    Ingrese tiempo de JugadorC: 13
Datos pista Pista2
    Ingrese tiempo de JugadorA: 30
    Ingrese tiempo de JugadorB: 33
    Ingrese tiempo de JugadorC: 30
Datos pista Pista3
    Ingrese tiempo de JugadorA: 0
    Ingrese tiempo de JugadorB: 7
    Ingrese tiempo de JugadorC: 17

Menor tiempo por pista
Pista1
    JugadorA (12.0)
Pista2
    JugadorA (30.0)
    JugadorC (30.0)
Pista3
    JugadorB (7.0)

Jugador con mejor tiempo promedio (20.0)
JugadorB
JugadorC

Mayor tiempo: 33.0
Pista2 : JugadorB
```

Observaciones: El archivo .py del problema se debe subir a Educa, en la sección "Trabajos". La plataforma Educa se cierra automáticamente. No hay entregas pasada la hora de término indicada en Educa. La hora oficial es la que dice Educa, **NO** la que dice el computador. Para asegurarme que leyó completamente estas instrucciones, agregue al código un comentario que contenga su nombre completo dentro de las primeras cinco líneas. Si el código no contiene el comentario, no será revisado. Una prueba respondida correctamente en un 60% corresponde a una nota 4.0.

Compromiso de honestidad

Mediante esta firma me comprometo a ser honesto al realizar esta evaluación, y a que entregaré el resultado de mi trabajo personal. (Al que no firma, no se le revisará la prueba)

Firma

Problema 1. Escriba acá la tabla con los valores de las variables

Problema 3. Dibuje acá la estructura de datos que va a utilizar, indicando qué almacenará en cada lista y matriz:



Prueba 2 Programación – 2016-1
Duración: 3 horas

20 Junio 2016

Solucion1:

Prueba 2 Programación – 2016-1
Duración: 3 horas

20 Junio 2016

Solución2:

```
# -*- coding: utf-8 -*-
```

```
import numpy as np
```

```
A = [  
    [1, -1, 0, 0],  
    [1, 0, 0, 0],  
    [0, 0, 1, -1],  
    [0, 1, -1, 0]  
]
```

```
print("a\tBomb\tSalida")  
print("-----")
```

```
for a in range(0, 11, 1):  
    b = 2 * a  
    B = [ 10-a, b+10, b-20, -10]
```

```
X = np.linalg.solve(A, B)
```

```
print(str(a) + "\t" + str(X[1]) + "\t" + str(X[3]))
```

```
#Z = np.dot(A, X)  
#print(Z)  
#print(B)
```

Prueba 2 Programación – 2016-1
Duración: 3 horas

20 Junio 2016

Solución3:

```
# -*- coding: utf-8 -*-
```

```
import numpy as np
```

```
nombres = []  
pistas = []
```

```
nombre = input("Ingrese nombre: ")  
while nombre != "FIN":  
    nombres.append(nombre)  
    nombre = input("Ingrese nombre: ")
```

```
pista = input("Ingrese pista: ")  
while pista != "FIN":  
    pistas.append(pista)  
    pista = input("Ingrese pista: ")
```

```
datos = np.zeros([len(pistas), len(nombres)])  
resumen = np.zeros([3, len(nombres)])
```

```
for p in range(len(pistas)):  
    print("Datos pista " + pistas[p])  
  
    for n in range(len(nombres)):  
        tiempo = float(input("\tIngrese tiempo de " + nombres[n] + ": "))  
        if tiempo > 0:  
            datos[p][n] = tiempo  
            resumen[0][n] = resumen[0][n] + tiempo # acumulamos  
            resumen[1][n] = resumen[1][n] + 1      # contamos  
            resumen[2][n] = resumen[0][n] / resumen[1][n] # promedio
```

```
# Por cada pista, el menor tiempo  
#
```

```
print("Menor tiempo por pista")  
for p in range(len(pistas)):  
    print(pistas[p])
```

```
menorTiempo = 999999
```

```
for n in range(len(nombres)):  
    if datos[p][n] != 0:  
        if datos[p][n] < menorTiempo:  
            menorTiempo = datos[p][n]
```

```
for n in range(len(nombres)):  
    if menorTiempo == datos[p][n]:  
        print("\t" + nombres[n] + " (" + str(menorTiempo) + ")")
```

```
# Ahora, el jugador con promedio más bajo  
#
```

```
menorTiempo = 999999  
for n in range(len(nombres)):  
    if resumen[1][n] > 0: # Sólo si el jugador compitió  
        if resumen[2][n] < menorTiempo:  
            menorTiempo = resumen[2][n]
```

```
print("Jugador con mejor tiempo promedio (" + str(menorTiempo) + ")")  
for n in range(len(nombres)):  
    if resumen[2][n] == menorTiempo:  
        print("\t" + nombres[n])
```

```
# Ahora, el menor tiempo entre todos  
# (Aunque esto se podría agregar al ciclo de lectura)
```

```
mayorTiempo = 0
```

```
for p in range(len(pistas)):  
    for n in range(len(nombres)):  
        if datos[p][n] != 0:
```


Prueba 2 Programación – 2016-1

Duración: 3 horas

20 Junio 2016

```
    if datos[p][n] > mayorTiempo:
        mayorTiempo = datos[p][n]

print("Mayor tiempo: " + str(mayorTiempo))

for p in range(len(pistas)):
    for n in range(len(nombres)):
        if datos[p][n] == mayorTiempo:
            print("\t" + pistas[p] + " : " + nombres[n])
```