Se dispone de un **arreglo de NumPy** que representa el **tiempo** de navegación en Internet de una empresa expresada en un *arreglo* de *n*x*m.*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *Sitios* | | | | | | |
| Empleados |  | *Sitios de Trabajo* | | | *Sitios de No Trabajo* | | |
| *Quipux* | *Espol* | *Mail* | *Twitter* | *Facebook* | *Tumblr* |
| *Leonardo* | ***130*** | ***91*** | ***83*** | ***10*** | ***32*** | ***0*** |
| *Jorge* | ***40*** | ***59*** | ***100*** | ***10*** | ***21*** | ***15*** |
| *Galo* | ***30*** | ***49*** | ***18*** | ***38*** | ***47*** | ***0*** |

Se dispone de las listas **empleados,** la lista de sitios de **trabajo**, y la lista de sitios de **No Trabajo.** La empresa requiere elaborar un reporte de las visitas a internet que incluya los siguientes datos:

a. Tiempo total de **uso de Internet.**

b. Tiempo total de uso de Internet **por empleado.**

c. Tiempo total de visita **por sitio.**

d. Tiempo total de visita **por** sitio de ***trabajo.***

e. Tiempo total de visita **por** sitios que ***no son de trabajo.***

f. El nombre del **empleado** que **más tiempo** ha pasado en sitios que **no son de trabajo.**

g. El sitio de **trabajo** que **más tiempo** ha sido usado.

h. Si el proveedor de Internet cobra un valor de *5 centavos* por minuto de visita a los sitios de *trabajo* y el *doble* para los *otros sitios.* Calcule el total a **pagar** en dólares.

i. ¿Cuántos **empleados** han visitado cada **sitio**?

j. Pídale al usuario el nombre de un empleado y muéstrele por pantalla un reporte que le diga cuanto tiempo el empleado ha pasado en cada sitio de trabajo (ordenados alfabéticamente).

k. Muestre el nombre de los empleados que han pasado más de 20 horas en Facebook.

l. Los empleados del literal k. le han pedido ayuda para hackear el sistema y reducir los minutos que han pasado en **todas** las páginas de **no trabajo** a la mitad de las que poseen actualmente.

m. Pídale un nombre de algún empleado al usuario, y elimínelo de la lista de empleados y de la matriz.

Desarrolle un programa en Python que calcule los resultados a las preguntas anteriores y muestre por pantalla las respuestas. Suponga que usted tiene una función llamada generarReporte( ), la cual crea el reporte como un arreglo Numpy (no se preocupe por cómo funciona esto, simplemente úselo de la siguiente manera para crear el arreglo numpy: arreglo= generarReporte( ))

**Solución:**

Tips:

1. Recuerde que el arreglo numpy son solo los NUMEROS. Los nombres de los sitios y el nombre de los empleados son solo para motivos de representación. Un arreglo no puede contener al mismo tiempo diferentes tipos de datos.
2. Recuerde utilizar funciones de numpy. Estas nos facilitan las soluciones en gran manera, a diferencia de tratar el arreglo como si fueran listas. Esto seria todo un dolor de cabeza.

import numpy as np  
empleados = []  
sitios\_trabajo = []  
sitios\_no\_trabajo = []  
arreglo = generarReporte()

# a. Se suman TODOS los elementos del arreglo.  
tiempo\_total = arreglo.sum()

# b. Obtengo la suma por cada una de las filas (empleado) de la matriz.   
tiempo\_por\_empleado = arreglo.sum(axis=1)

# c. Obtengo la suma por cada una de las columnas (sitios) de la matriz.  
tiempo\_por\_sitio = arreglo.sum(axis=0)

# d. Ya tengo el tiempo total de visita por cada uno los sitios en b. Y sabemos que los sitios de trabajo son las primeras N columnas del arreglo. Donde N es el número de sitios de trabajo que existen, lo cual lo puedo calcular obteniendo la longitud de la lista de sitios de trabajo. Por lo tanto, a partir del arreglo obtenido en b, utilizando slicing, obtengo los tiempos por sitio de trabajo.   
tiempo\_por\_sitio\_trabajo = tiempo\_por\_sitio[0:len(sitios\_trabajo)]

# e. Sabemos que los sitios de no trabajo, son las ultimas N columnas del arreglo. Utilizando una lógica similar a la usada en c. Voy a hacer un slicing desde donde terminan los sitios de trabajo, hasta el final del arreglo. Lo cual, es la suma de tiempo correspondiente a los sitios de no trabajo.  
tiempo\_por\_sitios\_no\_trabajo = tiempo\_por\_sitio[len(sitios\_trabajo):]

# f. Primero tengo que particionar el arreglo inicial, solo con los sitios de no trabajo. Luego, obtengo la suma de tiempo que cada empleado ha pasado en estos sitios, haciendo una suma por filas (axis = 1). Finalmente obtengo la posicion del elemento maximo, esta posicion me representa la fila que ha pasado mas tiempo en sitios de no trabajo, y las filas me representan empleados, cuyos nombres tengo en mi lista empleados.  
arreglo\_no\_trabajo = arreglo[:, len(sitios\_trabajo:]  
tiempo\_por\_empleado\_sitios\_nt = arreglo\_no\_trabajo.sum(axis=1)  
posicion\_empleado\_mas\_vago = tiempo\_por\_empleado\_sitios\_nt.argmax()  
empleado\_mas\_vago = empleados[posicion\_empleado\_mas\_vago]

# g.   
posicion\_sitio\_mas\_visitado = tiempo\_por\_sitio\_trabajo.argmax()  
sitio\_vas\_visitado = sitios\_trabajo[posicion\_sitio\_mas\_visitado]

# h. Utilizando las respuestas de d. y e., sumo todos los elementos de estos arreglos y obtengo el tiempo total de visita en sitios de trabajo y sitios de no trabajo. Multiplico este valor total por el multiplicador que nos dice el ejercicio y finalmente los sumo.  
valor\_trabajo = tiempo\_por\_sitio\_trabajo.sum() \* 0.05  
valor\_no\_trabajo = tiempo\_por\_sitios\_no\_trabajo.sum() \* 0.10  
valor\_a\_pagar = valor\_trabajo + valor\_no\_trabajo

# i. Sabemos que un empleado ha visitado un sitio, cuando este ha pasado más de cero minutos en el mismo. Por lo tanto, lo que tenemos que hacer es iterar el arreglo por columnas (sitios). En cada iteración del lazo for, obtengo la columna correspondiente a cada sitio utilizando slicing. Luego utilizando indexación booleana, en esa columna obtengo todos los elementos que sean mayores que cero, es decir, todos los empleados que hayan pasado mas de Cero minutos en esa pagina. Obteniendo la dimensión de ese arreglo (len), tengo el numero de empleados que han visitado ese sitio.  
empleados\_por\_sitio = []  
sitios = sitios\_trabajo + sitios\_no\_trabajo  
for i in range(0, len(sitios)):  
 t\_de\_sitio = arreglo[:, i]  
 numero\_empleados = len(t\_de\_sitio[t\_de\_sitio>0])  
 empleados\_por\_sitio.append(numero\_empleados)