

FUNDAMENTOS DE PROGRAMACIÓN EN PYTHON

[CC-BY-NC-SA] Leonardo Marco

PPP-PE-P10. Estructuras de datos

ALUMNO: Alejandro Bermúdez Gorozabel

- 1) Crea una lista `macs` con las direcciones MAC del aula:

70:85:c2:63:6a:5b
70:85:c2:76:a9:1b
70:85:c2:63:69:0f
70:85:c2:76:a9:1d
70:85:c2:76:a7:97
70:85:c2:8b:b9:b9
a8:a1:59:13:d4:22
d8:bb:c1:42:b9:db
70:85:c2:8b:b9:b7
70:85:c2:76:a8:fb
70:85:c2:76:a6:3e
70:85:c2:76:a6:36
70:85:c2:76:a9:23
70:85:c2:76:aa:04
70:85:c2:76:a5:80
70:85:c2:76:a4:a7
70:85:c2:76:a8:df
70:85:c2:93:45:3f
70:85:c2:76:a6:5b
70:85:c2:76:a9:f8
70:85:c2:8b:b9:bd
70:85:c2:76:a9:c4
d8:50:e6:48:f9:e1
d8:50:e6:48:f9:cc
d8:50:e6:48:f9:c8
70:85:c2:93:3a:9f

1. Añade la MAC 70:85:c2:93:55:f6 en la última posición.
2. Añade la MAC 70:85:c2:63:6c:49 en la tercera posición desplazando a las demás posiciones.
3. Modifica el valor de la penúltima posición por d8:50:e6:48:f9:c8.

4. Crea una nueva lista por compresión `macs_win` en la que se cambie el formato de cada MAC a formato Windows: DD-DD-DD-DD-DD-DD (todo mayúsculas y separador -).
5. Crea una nueva lista `macs_filas` utilizando extracción de elementos en la que:
 - Su primer elemento sea una lista con los primeros 4 elementos de la original en orden inverso.
 - Su segundo elemento sea una lista con los siguientes 6 elementos de la original en orden inverso.
 - Su tercero elemento sea una lista con los siguientes 6 elementos de la original en orden inverso.
 - Su cuarto elemento sea una lista con los siguientes 6 elementos de la original en orden inverso.
 - Su quinto elemento sea una lista con los siguientes 6 elementos de la original en orden inverso.
6. Accede al último elemento de `macs`
7. Accede al antepenúltimo elemento de `macs`.
8. Accede a la tercera MAC de la segunda fila de `macs_filas`.
9. Realiza una copia superficial de `macs` en `macs_cs`.
10. Modifica una MAC de `macs_cs` y comprueba si han cambiado en ambas listas.
11. Realiza una copia superficial de `macs_filas` en `macs_filas_cs` y otra profunda en `macs_filas_cp`.
12. Modifica una MAC de `macs_filas_cs` y otra de `macs_filas_cp` y comprueba si ha cambiado la lista `macs_filas`.
13. Ordena la lista `macs`. Comprueba si has modificado la lista original o has creado una lista nueva ordenada.

```
import copy

macs = [
    "70:85:c2:63:6a:5b",
    "70:85:c2:76:a9:1b",
    "70:85:c2:63:69:0f",
    "70:85:c2:76:a9:1d",
    "70:85:c2:76:a7:97",
    "70:85:c2:8b:b9:b9",
```

```
"a8:a1:59:13:d4:22",  
"d8:bb:c1:42:b9:db",  
"70:85:c2:8b:b9:b7",  
"70:85:c2:76:a8:fb",  
"70:85:c2:76:a6:3e",  
"70:85:c2:76:a6:36",  
"70:85:c2:76:a9:23",  
"70:85:c2:76:aa:04",  
"70:85:c2:76:a5:80",  
"70:85:c2:76:a4:a7",  
"70:85:c2:76:a8:df",  
"70:85:c2:93:45:3f",  
"70:85:c2:76:a6:5b",  
"70:85:c2:76:a9:f8",  
"70:85:c2:8b:b9:bd",  
"70:85:c2:76:a9:c4",  
"d8:50:e6:48:f9:e1",  
"d8:50:e6:48:f9:cc",  
"d8:50:e6:48:f9:c8",  
"70:85:c2:93:3a:9f"  
]
```

```
macs.append("70:85:c2:93:55:f6")

macs.insert(2, "70:85:c2:63:6c:49")

macs[-2] = "d8:50:e6:48:f9:c8"

macs_win = [mac.replace(":", "-").upper() for mac in macs]

macs_filas = [
    macs[0:4][::-1],
    macs[4:10][::-1],
    macs[10:16][::-1],
    macs[16:22][::-1],
    macs[22:28][::-1]
]

ultimo = macs[-1]
print(ultimo)

antepenultimo = macs[-3]
```

```
print(antepenultimo)

tercera_segunda_fila = macs_filas[1][2]

print(tercera_segunda_fila)

macs_cs = macs.copy()

macs[0] = "00:00:00:00:00:00"

#cambia

print(macs[0])

#no cambia

print(macs_cs[0])

macs_filas_cs = macs_filas.copy()

macs_filas_cp = copy.deepcopy(macs_filas)

macs_filas_cs[0][0] = "AA:AA:AA:AA:AA:AA"

macs_filas_cp[1][0] = "BB:BB:BB:BB:BB:BB"

# CAMBIA (copia superficial)

print(macs_filas[0][0])
```

```

# NO CAMBIA (copia profunda)

print(macs_filas[1][0])

macs.sort()

print(macs)

#se modifica la original

#para no modificarla y crear una nueva:

macs_ordenada = sorted(macs)

```

2) Crea un diccionario anidado `macs` en el que se almacenen las direcciones MAC del aula en dos dimensiones, de forma que la primera dimensión concuerde con la fila del equipo y la segunda con la columna. Por ejemplo `macs[2][3]` sea la MAC del equipo 23.

- Muestra la MAC del equipo 33.
- Crea un nuevo diccionario por compresión `macs_win` en el que se cambie el formato de cada MAC a formato Windows: DD-DD-DD-DD-DD-DD (todo mayúsculas y separador -).

```

11 = 70:85:c2:63:6a:5b
12 = 70:85:c2:76:a9:1b
13 = 70:85:c2:63:69:0f
14 = 70:85:c2:63:6c:49
21 = 70:85:c2:76:a9:1d
22 = 70:85:c2:76:a7:97
23 = 70:85:c2:8b:b9:b9
24 = a8:a1:59:13:d4:22
25 = d8:bb:c1:42:b9:db
26 = 70:85:c2:8b:b9:b7
31 = 70:85:c2:76:a8:fb
32 = 70:85:c2:76:a6:3e
33 = 70:85:c2:76:a6:36

```

```
34 = 70:85:c2:76:a9:23
35 = 70:85:c2:76:aa:04
36 = 70:85:c2:76:a5:80
41 = 70:85:c2:76:a4:a7
42 = 70:85:c2:76:a8:df
43 = 70:85:c2:93:45:3f
44 = 70:85:c2:76:a6:5b
45 = 70:85:c2:76:a9:f8
46 = 70:85:c2:8b:b9:bd
51 = 70:85:c2:76:a9:c4
52 = d8:50:e6:48:f9:e1
53 = d8:50:e6:48:f9:cc
54 = d8:50:e6:48:f9:c9
55 = 70:85:c2:93:3a:9f
56 = 70:85:c2:93:55:f6
```

```
macs = {

  1: {

    1: "70:85:c2:63:6a:5b",

    2: "70:85:c2:76:a9:1b",

    3: "70:85:c2:63:69:0f",

    4: "70:85:c2:63:6c:49"

  },

  2: {

    1: "70:85:c2:76:a9:1d",

    2: "70:85:c2:76:a7:97",

    3: "70:85:c2:8b:b9:b9",

    4: "a8:a1:59:13:d4:22",

    5: "d8:bb:c1:42:b9:db",

  }

}
```

```
6: "70:85:c2:8b:b9:b7"

},

3: {

    1: "70:85:c2:76:a8:fb",

    2: "70:85:c2:76:a6:3e",

    3: "70:85:c2:76:a6:36",

    4: "70:85:c2:76:a9:23",

    5: "70:85:c2:76:aa:04",

    6: "70:85:c2:76:a5:80"

},

4: {

    1: "70:85:c2:76:a4:a7",

    2: "70:85:c2:76:a8:df",

    3: "70:85:c2:93:45:3f",

    4: "70:85:c2:76:a6:5b",

    5: "70:85:c2:76:a9:f8",

    6: "70:85:c2:8b:b9:bd"

},

5: {

    1: "70:85:c2:76:a9:c4",

    2: "d8:50:e6:48:f9:e1",
```

```

    3: "d8:50:e6:48:f9:cc",
    4: "d8:50:e6:48:f9:c9",
    5: "70:85:c2:93:3a:9f",
    6: "70:85:c2:93:55:f6"

}

print(macs[3][3])

macs_win = {

    fila: {

        col: mac.replace(":", "-").upper()

        for col, mac in columnas.items()

    }

    for fila, columnas in macs.items()

}

print(macs_win[3][3])

```

- 3) El club deportivo *FitLife* ofrece 3 actividades: musculación, crossfit y hyrox. El club tiene los siguientes socios actualmente:

```

musculacion = {"Ana", "Luis", "Carlos", "María", "Sofía"}
crossfit = {"María", "Jorge", "Sofía", "Lucía", "Carlos"}
hyrox = {"Luis", "Jorge", "Pedro"}

```

1. Muestra la lista de socios de cada actividad.
 2. Calcula el número total de socios únicos y muestra la lista de socios únicos.
 3. Muestra los socios que practican tanto musculación como crossfit.
 4. Muestra los socios que practica únicamente una actividad.
 5. Muestra los socios que no practican hyrox.
 6. Muestra los socios que practica más de una actividad.
 7. Agrega un socio Raúl a hyrox.
 8. Elimina a Lucía de crossfit (si existe).

```
solo_crossfit = crossfit - musculacion - hyrox

solo_hyrox = hyrox - musculacion - crossfit

solo_una = solo_musculacion | solo_crossfit | solo_hyrox

print("Socios que practican solo una actividad:", solo_una)

sociosNo_hyrox = (musculacion | crossfit) - hyrox

print("Socios que no practican hyrox:", sociosNo_hyrox)

sociosMas_de_una = (

    (musculacion & crossfit) |

    (musculacion & hyrox) |

    (crossfit & hyrox)

)

print("Socios que practican más de una actividad:", sociosMas_de_una)

hyrox.add("Raúl")

print("Hyrox con Raúl:", hyrox)

#discard da error si no existe

crossfit.discard("Lucía")

print("Crossfit sin Lucía:", crossfit)
```