

# Alternativa I - Solucion posible del parcial de fecha 9/10/2018

# punto 1 - Alternativa I

```
def calcular_temp_promedio(dic_datos):  
    l_temp_prom = []  
    for clave in dic_datos:  
        t_min = 0  
        t_max = 0  
        for dias in dic_datos[clave]:  
            t_min += dias[1]  
            t_max += dias[2]  
        l_temp_prom.append([clave, round(t_min/len(dias),1), round(t_max/len(dias),1)])  
    l_temp_prom.sort()  
    return l_temp_prom
```

# punto 2

```
def min_tmax_mes(dic_datos):  
    min_max = min([dias[2] for clave in dic_datos for dias in dic_datos[clave]])  
    return min_max
```

# punto 2 - Alternativa I

```
def ciudades_con_min_max(dic_datos, min_max):  
    l_ciudades = []  
    for ciudad in dic_datos:  
        l = []  
        for dia in dic_datos[ciudad]:  
            l.append(dia[2])  
        if min_max in l:  
            l_ciudades.append(ciudad)  
    return l_ciudades
```

# punto 3 - Alternativa I

```
def temp_min_prom(dic_datos):  
    tmin_prom = 0  
    for ciudad in dic_datos:  
        tmin_prom += dic_datos[ciudad][-1][1]  
    return tmin_prom/len(dic_datos)
```

# punto 4 - Alternativa I

```
def mayor_amplitud(dic_datos):  
    l = []  
    for clave in dic_datos:  
        ldias = []  
        max_at = dic_datos[clave][0][2] - dic_datos[clave][0][1]  
        for dias in dic_datos[clave]:  
            if (dias[2] - dias[1]) > max_at:  
                max_at = dias[2] - dias[1]  
                ldias = [dias[0]]  
            elif dias[2]-dias[1] == max_at:  
                ldias.append(dias[0])  
        l.append([clave, max_at, ldias])  
    l.sort(key=lambda x:x[1])  
    return l
```

# punto 5

```
def mostrar_datos_ciudad(dic_datos):  
    print("\nDatos de Ciudades")  
    ciudad = input("Ciudad: ")  
    while ciudad:  
        if ciudad in dic_datos:  
            print("Ciudad: ", ciudad)  
            print("\tDia \tTemp Min \tTemp Max")  
            for dia in dic_datos[ciudad]:  
                print("\t{0:3d} \t{0:5d} \t{0:5d}".format(dia[0], dia[1], dia[2]))
```

```

        else:
            print("Ciudad Inexistente")
            ciudad = input("Ciudad: ")
        return

#----- Bloque Principal -----#

import datos
dic_temp = datos.cargar_datos(10, 2018)

# Punto 1
temp_prom_por_ciudad = calcular_temp_promedio(dic_temp)
for ciudad in temp_prom_por_ciudad:
    print(ciudad[0].ljust(20), ciudad[1], ciudad[2], sep="\t ")

# Punto 2
tmin_max = min_tmax_mes(dic_temp)
print("\nMenor Temperatura Maxima del Mes: ", tmin_max)
print("Ciudades: ", ciudades_con_min_max(dic_temp, tmin_max))

# Punto 3
print("\nTemp. min. promedio el ultimo dia del mes: ", temp_min_prom(dic_temp))

# Punto 4
print("\nMayor Amplitud Termica por Ciudad")
l_amp_term = mayor_amplitud(dic_temp)
for ciudad in l_amp_term:
    print("Ciudad: {0:20} Amplitud: {1:2d}    Dias: {2}".format(ciudad[0], ciudad[1],
        ciudad[2]))

# Punto 5
mostrar_datos_ciudad(dic_temp)

```

2