

```
'''
```

Para realizar una estadística la empresa "El estadista", le pide desarrollar un sistema que procese los datos solicitados en una encuesta. Para ello se carga la edad, cantidad de hijos y la provincia donde vive (se codifican de 1 a 10).

La carga finaliza cuando se ingresa 0 en la edad o se hayan procesados mas de 1000 encuestas

A partir de estos datos cargados se pide resolver las siguiente estadísticas:

A) Mostrar todos los datos de los encuestados en forma de cuadro.

B) Porcentaje de encuestados de cada provincia.

C) Indicar la edad y la provincia de la persona encuestada que tenga menos hijos.

D) Promedio de edades de los encuestados de una provincia en particular (el dato de la provincia se debe pedir por teclado).

E) Mostrar todos los datos de los encuestados ordenado por provincia en forma descendente.

```
'''
```

```
# FUNCIONES
```

```
def validar_provincia():
```

```
    provincia_ent = int(input("Ingrese su provincia (1 al 10): "))
```

```
    while provincia_ent <= 0 or provincia_ent > 10:
```

```
        provincia_ent = int(input("ERROR. Ingrese su provincia (1 al 10): "))
```

```
    return provincia_ent
```

```
def validar_edad():
```

```
    edad_ent = int(input("\nIngrese su edad (0 para finalizar): "))
```

```
    while edad_ent < 0:
```

```
        edad_ent = int(input("ERROR. Ingrese su edad (0 para finalizar): "))
```

```
    return edad_ent
```

```
def validar_hijos():
```

```
    cant_hijos = int(input("Ingrese la cantidad de hijos: "))
```

```
    while cant_hijos < 0:
```

```
        cant_hijos = int(input("ERROR. Ingrese la cantidad de hijos: "))
```

```
    return cant_hijos
```

```
def cargar_datos(arr_edades, arr_hijos, arr_provincias):
```

```
    edad = validar_edad()
```

```
    while edad != 0 and len(arr_edades) < 1000:
```

```
        cant_hijos = validar_hijos()
```

```
        provincia = validar_provincia()
```

```
        arr_edades.append(edad)
```

```
        arr_hijos.append(cant_hijos)
```

```
        arr_provincias.append(provincia)
```

```
        edad = validar_edad()
```

```
    print("\nCarga de datos finalizada.")
```

```
def mostrar_datos(arr_edades, arr_hijos, arr_provincias):
```

```
    print(f"Edad\t Hijos\t Provincia\t")
```

```
    print("-----")
```

```
    for i in range (len(arr_edades)):
```

```
        print(f"{arr_edades[i]}\t {arr_hijos[i]}\t {arr_provincias[i]}\t")
```

```
def min_hijos (arr_hijos):
```

```
    posMin = 0
```

```
    for i in range (len(arr_hijos)):
```

```

        if arr_hijos[i] < arr_hijos[posMin]:
            posMin = i
    return posMin

def porcentajes(arr_provincias, provincia):
    acum_por = 0
    cant = 0
    for i in range(len(arr_provincias)):
        if arr_provincias[i] == provincia:
            acum_por += 1
            cant += 1

    cant = len(arr_provincias)
    if cant > 0:
        porcentaje = (acum_por / cant) * 100
    else:
        porcentaje = 0.0 # Si no hay nadie, el porcentaje es 0
    return porcentaje

def calcular_porcentajes(arr_provincias):
    print(f"Provincia\t Porcentaje")
    print("-----")
    for i in range(10):
        print(f"{i + 1}\t\t {porcentajes(arr_provincias, i + 1):.2f}")

def promedio_edades(arr_edades, arr_provincias):
    provincia = validar_provincia()
    acum = 0
    cont = 0
    for i in range (len(arr_edades)):
        if arr_provincias[i] == provincia:
            cont += 1
            acum += edades[i]
    promedio = acum/cont
    if cont != 0:
        print(f"\nEl promedio de edades de la provincia {provincia} es:")
        print(f"{promedio:.2F}")
    else:
        print("\nLa provincia ingresada no se encuentra en la lista.")

def intercambiar (arreglo, i, j):
    aux = arreglo[i]
    arreglo[i] = arreglo[j]
    arreglo[j] = aux

def ordenar_datos (arr_provincias, arr_edades, arr_hijos):
    for i in range (0, len(arr_provincias)-1):
        for j in range (i + 1, len(arr_provincias)):
            if arr_provincias[i] < arr_provincias[j]:
                intercambiar(arr_provincias,i,j)
                intercambiar(arr_edades,i,j)
                intercambiar(arr_hijos,i,j)

# PPAL
edades = []

```

```

hijos_cant = []
provincias = []

print("\nCargar datos:")
cargar_datos(edades, hijos_cant, provincias)
print("-----")
if len(edades) != 0:
    # A) Mostrar todos los datos de los encuestados en forma de cuadro.
    print("\nA) Mostrar los datos:")
    print("\nDATOS CARGADOS...\n")
    mostrar_datos(edades, hijos_cant, provincias)
    print("-----")
    # B) Porcentaje de encuestados de cada provincia.
    print("\nB) Porcentaje de cada provincia:\n")
    calcular_porcentajes(provincias)
    print("-----")
    # C) Indicar la edad y la provincia de la persona encuestada que tenga menos hijos.
    print("\nC) Persona encuestada con menos hijos:")
    menos_hijos = min_hijos(hijos_cant)
    print(f"\nCantidad de hijos {hijos_cant[menos_hijos]}, edad {edades[menos_hijos]},
provincia {provincias[menos_hijos]}.")
    print("-----")
    # D) Promedio de edades de los encuestados de una provincia en particular (el dato de
la provincia se debe pedir por teclado).
    print("\nD) Promedio de edades en una provincia:\n")
    promedio_edades(edades, provincias)
    print("-----")
    #E) Mostrar todos los datos de los encuestados ordenado por provincia en forma
descendente.
    print("\nE) Datos ordenados por:\n")
    print("PROVINCIA EN ORDEN DESCENDENTE\n")
    ordenar_datos(provincias, edades, hijos_cant)
    mostrar_datos(edades, hijos_cant, provincias)
    print("-----")
print("\nFin del programa.\n")

```