

EJERCICIO 1:

Una barrio privado posee una lista de los inmuebles que tiene en alquiler con los siguientes datos:

- Nro. Lote de la casa
- Nombre del titular
- Cantidad de ambientes (1 – 2 – 3 – 4)
- Precio del alquiler anual

- 1) Diseñar un programa que permita cargar toda la lista en cuatro arreglos: LOTES, TITULARES, AMBIENTES y ALQUILER. La carga finaliza con lote = 0.
- 2) Listar los datos en cuatro columnas LOTE - TITULAR - AMBIENTES - ALQUILER
- 3) Calcular el precio promedio de las casas del barrio privado.
- 4) Determinar de cuántos ambientes tiene el alquiler más caro

```
def entero_positivo(mensaje):
```

```
    num=int(input(mensaje))
    while(num<0):
        num=int(input(mensaje))
    return num
```

```
def numero_positivo_entre(mensaje,minimo,maximo):
```

```
    num=int(input(mensaje))
    while(num<minimo or num>maximo):
        num=int(input("ERROR. Reingrese cantidad de ambientes:"))
    return num
```

```
def validar_texto(mensaje):
```

```
    texto=input(mensaje)
    while(texto==""):
        texto=input("ERROR NOMBRE VACIO. Reingrese:")
    return texto
```

```
def cargar_vectores(vec_lote,vec_titular,vec_ambiente,vec_alquiler):
```

```
    nrolote=entero_positivo("Ingresa numero de lote:")
    while not(nrolote==0):
        vec_lote.append(nrolote)
        titular=validar_texto("Ingresa nombre del titular:")
        vec_titular.append(titular)
        nroambientes=numero_positivo_entre("Ingresa cantidad de ambientes:",1,4)
        vec_ambiente.append(nroambientes)
        precio_alquiler=entero_positivo("Ingresa precio del alquiler anual:")
        vec_alquiler.append(precio_alquiler)
```

```

nrolote=entero_positivo("Ingresa numero de lote:")

def imprimir_lista(vec_lote,vec_titular,vec_ambiente,vec_alquiler):
    print("Listado de los datos")
    print("lote-titular-ambientes-alquiler anual")
    for i in range(len(vec_lote)):
        print(vec_lote[i],"--",vec_titular[i],"--",vec_ambiente[i],"--",vec_alquiler[i])

def promedio(vec_alquiler):
    suma=0
    cant=0
    for i in range(len(vec_alquiler)):
        suma = suma + vec_alquiler[i]
        cant= cant + 1
    if(cant>0):
        promedio=suma/cant
    print("Promedio de las casas del barrio privado es de:",promedio)

def determinar_alquiler_mayor(vec_ambiente,vec_alquiler):
    cant_mayor_alquiler=0
    for i in range(len(vec_ambiente)):
        if(vec_alquiler[i]>cant_mayor_alquiler):
            cant_mayor_alquiler=vec_alquiler[i]
            cant_ambiente=vec_ambiente[i]
    print("La cantidad de ambientes mas costosa es:",cant_ambiente)

lotes=[]
titulares=[]
ambientes=[]
alquiler=[]
cargar_vectores(lotes,titulares,ambientes,alquiler)
imprimir_lista(lotes,titulares,ambientes,alquiler)
promedio(alquiler)
determinar_alquiler_mayor(ambientes,alquiler)

```

EJERCICIO 2:

Una empresa de Viajes de Egresados guarda la información de los alumnos que quieren viajar: - **nombre y apellido** - **sexo** - **Destino:** Bariloche - Porto Seguro.

Ingresar los datos de todos los viajeros, hasta que no se quiera cargar más viajeros (ingresan "fin" en el nombre). Se pide:

- Separar mujeres y varones:** se pide guardar en un arreglo el nombre de las mujeres y en otro el de los hombres.
- Listar ambos grupos de datos ordenados alfabéticamente** para pasárselo a la empresa de micros. Agregar el número de viajero (empezando en 1 en adelante). Si alguno de los vectores quedará vacío, Indicarlo con un mensaje por pantalla.
- Mostrar el Porcentaje de Hombres y Mujeres:** Calcular e imprimir los porcentajes de alumnos que eligieron cada destino.
- Decir cual es el destino con mayor cantidad de viajeros

```
def entero_positivo(mensaje):
```

```
    num=int(input(mensaje))
    while(num<0):
        num=int(input(mensaje))
    return num
```

```
def validar_texto(mensaje):
```

```
    texto=input(mensaje)
    while(texto==""):
        texto=input("ERROR VACIO. Reingrese:")
    return texto
```

```
def cargar_vector(vec_nombre,vec_sexo,vec_destino):
```

```
    nombre=validar_texto("Ingresa nombre del alumno:").upper()
    nombre.upper()
    while not(nombre=="FIN"):
        apellido=validar_texto("Ingresa apellido del alumno:").upper()
        vec_nombre.append(nombre+"-"+apellido)
        sexo=input("Ingresa sexo del alumno:").upper()
        while not(sexo=="F" or sexo=="M"):
            sexo=input("Ingresa sexo del alumno:").upper()
        vec_sexo.append(sexo)
        destino=input("Ingresa destino:").upper()
        while not(destino=="BARILOCHE" or destino=="PORTO"):
            destino=input("Ingresa destino:").upper()
        vec_destino.append(destino)
        nombre=validar_texto("Ingresa nombre del alumno:").upper()
```

```
def generar_arreglo(vec_nombre,vec_sexo,vec_mujeres,vec_hombres):
    for i in range(len(vec_sexo)):
        if(vec_sexo[i]=="F" or vec_sexo[i]=="f"):
            vec_mujeres.append(vec_nombre[i])
        else:
            vec_hombres.append(vec_nombre[i])
```

```
def calcular_porcentaje(cant,cant_total):
    porcentaje=0
    if(cant_total>0):
        porcentaje=cant/cant_total*100
    return porcentaje
```

```
def calcular_cantidad(vector):
    cantidad=len(vector)
    return cantidad
```

```
def calcular_cant_destino(vec_destino,condicion):
    cant=0
    for i in range(len(vec_destino)):
        if(vec_destino[i]==condicion):
            cant =cant + 1
    return cant
```

```
def ordenar_vector(vec_nya):
    for i in range(1,len(vec_nya)):
        for j in range(0,len(vec_nya)-i):
            if(vec_nya[j]>vec_nya[j+1]):
                aux=vec_nya[j]
                vec_nya[j]=vec_nya[j+1]
                vec_nya[j+1]=aux
```

```
def imprimir_arreglo(vector):
    print("Grupos ordenados alfabéticamente")
    for i in range(len(vector)):
        print(i+1,vector[i])
```

```
nombre=[]
sexo=[]
destino=[]
mujeres=[]
hombres=[]
nyap=[]
cargar_vector(nombre,sexo,destino)
```

```
generar_arreglo(nombre,sexo,mujeres,hombres)
cantidad_mujeres= calcular_cantidad(mujeres)
cantidad_hombres= calcular_cantidad(hombres)
total=cantidad_hombres+cantidad_mujeres
porcentaje_mujeres=calcular_porcentaje(cantidad_mujeres,total)
print("El porcentaje de mujeres es:",porcentaje_mujeres)
porcentaje_hombres=calcular_porcentaje(cantidad_hombres,total)
print("El porcentaje de hombres es de:",porcentaje_hombres)
cantidad_bariloche=calcular_cant_destino(destino,"BARILOCHE")
cantidad_porto=calcular_cant_destino(destino,"PORTO")
porcentaje_bariloche=calcular_porcentaje(cantidad_bariloche,total)
print("El porcentaje de alumnos que viajaron a bariloche:",porcentaje_bariloche)
porcentaje_porto=calcular_porcentaje(cantidad_porto,total)
print("El porcentaje de alumnos que viajaron a Porto Seguro:",porcentaje_porto)
ordenar_vector(mujeres)
ordenar_vector(hombres)
imprimir_arreglo(mujeres)
imprimir_arreglo(hombres)
```

Ejercicio 3

Un oftalmólogo le gusta hacer estadísticas de sus pacientes al final del día. Guarda los siguientes datos de cada paciente:

- Obra social (1: Osde, 2: Swiss, 3: Italiano, 4: Otros)
- Nombre
- Edad

Te pide que le hagas un programa que le permita:

1. Ingresar los datos de los 10 pacientes que atiende en el día. Mostrarlos con los nombres de los pacientes en mayúscula y los nombres de las obras sociales en minúscula.
2. Informe la cantidad de pacientes mayores a 21 años y qué porcentaje representa del total de los pacientes atendidos.
3. Según el total de pacientes, calcular el porcentaje de pacientes según la obra social
4. Cuál es la obra social de la que se atendió más pacientes. Mostrar su nombre.

```
def entero_positivo(mensaje):  
    num=int(input(mensaje))  
    while(num<0):  
        num=int(input(mensaje))  
    return num
```

```
def numero_entre(mensaje,minimo,maximo):  
    numero=int(input(mensaje))  
    while(numero<minimo or numero>maximo):  
        numero=int(input(mensaje))  
    return numero
```

```
def validar_texto(mensaje):  
    texto=input(mensaje)  
    while(texto == ""):  
        texto=input("ERROR. Reingrese:")  
    return texto
```

```
def cargar_vector(vec_obra,vec_nombre,vec_edad):  
    for i in range(10):  
        vec_obra[i]=numero_entre("Ingresa obra social| 1-Osde|2-Swiss|3-Italiano|4-Otros:",1,4)  
        vec_nombre[i]=validar_texto("Ingresa nombre del paciente:").upper()
```

```
vec_edad[i]=entero_positivo("Ingresa edad del paciente:")
```

```
def imprimir_vectores(vec_obra,vec_nombre):
```

```
    for i in range(len(vec_obra)):
        if(vec_obra[i]==1):
            print(vec_nombre[i], "osde")
        elif(vec_obra[i]==2):
            print(vec_nombre[i], "swiss")
        elif(vec_obra[i]==3):
            print(vec_nombre[i], "italiano")
        else:
            print(vec_nombre[i], "otros")
```

```
def calcular_pacientes_mayores(vector):
```

```
    cant=0
    for i in range(len(vector)):
        if(vector[i]>21):
            cant=cant + 1
    return cant
```

```
def calcular_porcentaje(vector,cant):
```

```
    total=len(vector)
    if(total>0):
        porcentaje=cant/total * 100
    return porcentaje
```

```
def vector_aux(vec_obra,vec_cant):
```

```
    for i in range(len(vec_obra)):
        vec_cant[vec_obra[i]]+=1
```

```
def calcular_porcentaje_x_obra(vec_cant):
```

```
    total=10
    for i in range(1,len(vec_cant)):
        porcentaje= vec_cant[i]/total * 100
        print("El porcentaje de pacientes segun la obra numero "+str(i)+" ,es de:",porcentaje)
```

```
def calcular_mayor_obra(vec_cant):
```

```
    cant_mayor=0
    for i in range(len(vec_cant)):
        if(vec_cant[i]>cant_mayor):
            cant_mayor=vec_cant[i]
            tipo_obra = i
    if(tipo_obra==1):
        print("La obra de la cual se atendieron mas pacientes fue osde")
    elif(tipo_obra==2):
```

```
    print("La obra de la cual se atendieron mas pacientes fue swiss")
elif(tipo_obra==3):
    print("La obra de la cual se atendieron mas pacientes fue italiano")
else:
    print("La obra de la cual se atendieron mas pacientes fue otros")
```

```
print("La cantidad de pacientes atendidos fue", cant_mayor)
```

```
obra_social=[0]*10
nombre_paciente=[0]*10
edad_paciente=[0]*10
cantidad_x_obra=[0]*5
cargar_vector(obra_social,nombre_paciente,edad_paciente)
pacientes_mayores_21=calcular_pacientes_mayores(edad_paciente)
print("La cantidad de pacientes mayores a 21 años es:",pacientes_mayores_21)
porcentaje_mayores_21=calcular_porcentaje(edad_paciente,pacientes_mayores_21)
print("El porcentaje de los pacientes mayores a 21 años es:",porcentaje_mayores_21)
vector_aux(obra_social,cantidad_x_obra)
calcular_porcentaje_x_obra(cantidad_x_obra)
calcular_mayor_obra(cantidad_x_obra)

'''
print(obra_social)
print(nombre_paciente)
print(edad_paciente)
print(cantidad_x_obra)
```


Ejercicio 4

1. Pedir el peso, la altura y el sexo, de un estudiante. Determinar la cantidad de vitaminas que debe consumir ese estudiante, en base al siguiente criterio:

“Si son varones, y su estatura es mayor a 1.60, y su peso es mayor o igual a 150 lb, su dosis, será: 20% de la estatura y 80% de su peso.

De lo contrario, la dosis será la siguiente: 30% de la estatura y 70% de su peso. Si son mujeres, y su estatura es mayor de a 1.50 m su peso es mayor o igual a 130 lb, su dosis será: 25% de la estatura y 75% de su peso.

De lo contrario, la dosis será: 35% de la estatura y 65% de su peso. La dosis debe ser expresada en gramos.”

```
def numero_positivo(mensaje):
    num=float(input(mensaje))
    while(num<0):
        num=int(input(mensaje))
    return num
```

```
def cargar_vectores(vec_peso,vec_altura,vec_sexo):
    peso=numero_positivo("Ingresa peso en libras del estudiante:")
    while not(peso==0):
        vec_peso.append(peso)
        altura=numero_positivo("Ingresa altura del estudiante:")
        vec_altura.append(altura)
        sexo=input("Ingresa sexo | F-Femenino | M-Masculino:").upper()
        while not(sexo=="F" or sexo=="M"):
            sexo=input("Ingresa sexo | F-Femenino | M-Masculino:").upper()
        vec_sexo.append(sexo)
        peso=numero_positivo("Ingresa peso del estudiante:")
```

```
def calcular_dosis(vec_peso,vec_altura,vec_sexo):
    print("Segun sus valores:")
    print("Peso - Altura - Sexo")
    for i in range(len(vec_peso)):
        if(vec_sexo[i]=="F"):
            if(vec_altura[i]>1.60 and vec_peso[i]>=150):
                altura = 20 * vec_altura[i] / 100
                peso = 80 * vec_peso[i] / 100
                dosis = altura + peso
                print(vec_peso[i],",",vec_altura[i],",",vec_sexo[i])
                print("La dosis sera:",dosis)
            else:
                altura = 30 * vec_altura[i] / 100
```

```

    peso = 70 * vec_peso[i] / 100
    dosis = altura + peso
    print(vec_peso[i],",",vec_altura[i],",",vec_sexo[i])
    print("La dosis sera:",dosis)
else:
    if(vec_altura[i]>1.50 and vec_peso[i]>=130):
        altura = 25 * vec_altura[i] / 100
        peso = 75 * vec_peso[i] / 100
        dosis = altura + peso
        print(vec_peso[i],",",vec_altura[i],",",vec_sexo[i])
        print("La dosis sera:",dosis)
    else:
        altura = 35 * vec_altura[i] / 100
        peso = 65 * vec_peso[i] / 100
        dosis = altura + peso
        print(vec_peso[i],",",vec_altura[i],",",vec_sexo[i])
        print("La dosis sera:",dosis)
peso=[]
altura=[]
sexo=[]
cargar_vectores(peso,altura,sexo)
calcular_dosis(peso,altura,sexo)

```

Ejercicio 5

Se ingresan la cantidad de kilos vendidos de chocolate amargo y semiamargo (dos vectores) a lo largo de el mes de junio (30 días). El valor del kilo de chocolate amargo es \$3500 y el valor del kilo del chocolate semiamargo es de \$2400. Calcular y mostrar:

a) El día del mes donde se vendieron más kilos de chocolate amargo y el día del mes donde se vendieron la menor cantidad de chocolate semiamargo.

b) Insertar la recaudación del mes al comienzo del arreglo en cada tipo de chocolate

c) Eliminar las ventas menores a 3 kilos de cada tipo de chocolate.

```
def entero_positivo(mensaje):
    numero=int(input(mensaje))
    while(numero<0):
        numero=int(input("Reingrese:"))
    return numero
```

```
def cargar_vectores(vec_amargo,vec_semi):
    for i in range(len(vec_amargo)):
        vec_amargo[i]=entero_positivo("Ingresa el numero de Kilos de chocolate amargo vendidos en el dia"+str(i+1)+":")
        vec_semi[i]=entero_positivo("Ingresa el numero de Kilos de chocolate semiamargo vendidos en el dia"+str(i+1)+":")
```

```
def calcular_maximo(vec_amargo):
    cant_max=0
    dia_max=1
    for i in range(len(vec_amargo)):
        if(vec_amargo[i]>cant_max):
            cant_max=vec_amargo[i]
            dia_max= i+1
    print("El dia del mes donde se vendieron mas kilos de chocolate amargo fue:",dia_max)
```

```
def calcular_minimo(vec_semi):
    cant_minima=vec_semi[0]
    dia_min=1
    for i in range(len(vec_semi)):
        if(vec_semi[i]<cant_minima):
            dia_min= i+1
    print("El dia del mes donde se vendieron menos kilos de chocolate semi amargo fue:",dia_min)
```

```

def sumar_kilos(vector):
    suma=0
    for i in range(len(vector)):
        suma=suma + vector[i]
    return suma

def imprimir_arreglo(mensaje,vector):
    print(mensaje)
    for i in range(len(vector)):
        print(vector[i])

def vector_sin_numeros_menores3(vector,vec_sin):
    for i in range(len(vector)):
        if(vector[i]>3):
            numero=vector[i]
            vec_sin.append(numero)

amargo=[0]*30
semi_amargo=[0]*30
vector_sin_amargo=[]
vector_sin_semi=[]
cargar_vectores(amargo,semi_amargo)
calcular_maximo(amargo)
calcular_minimo(semi_amargo)
recaudacion_amargo=sumar_kilos(amargo)*3500
print("La recaudacion del mes del chocolate amargo fue:",recaudacion_amargo)
recaudacion_semi=sumar_kilos(semi_amargo)*2400
print(sumar_kilos(semi_amargo))
print("La recaudacion del mes del chocolate semiamargo fue:",recaudacion_semi)
amargo.insert(0,recaudacion_amargo)
imprimir_arreglo("El arreglo de chocolate amargo con el monto de la recaudacion en
primera posicion:",amargo)
semi_amargo.insert(0,recaudacion_semi)
imprimir_arreglo("El arreglo de chocolate semiamargo con el monto de la recaudacion en
primera posicion:",semi_amargo)
vector_sin_numeros_menores3(amargo,vector_sin_amargo)
vector_sin_numeros_menores3(semi_amargo,vector_sin_semi)
amargo=vector_sin_amargo.copy()
semi_amargo=vector_sin_semi.copy()
imprimir_arreglo("Vector de chocolate amargo sin las ventas menores a 3:",amargo)
imprimir_arreglo("Vector de chocolate semiamargo sin las ventas menores a
3:",semi_amargo)

```