

“Se nace lo que se es... O se será aquello lo que se crea...”

— Jorge Drexler.

Apellido, Nombre: ..... Padron: .....

Mail: ..... Cantidad de hojas entregadas: .....

Teórica		
1	2	3

Práctica			
1	2	3	4

Nota Final

### Aclaraciones:

- Antes de comenzar a resolver el examen, complete sus datos en esta hoja, y al finalizarlo, fírmela.
- Los ejercicios deben ser implementados en el lenguaje de programación C.
- Para cada ejercicio se recomienda fuertemente que realicen un análisis y un diagrama del problema y la solución.
- Se deben numerar TODAS las hojas e inicializarlas con nombre, apellido, padrón y cualquier otra información que considere necesaria.
- La aprobación del examen está sujeta a la correcta realización de al menos el 60 % del mismo.

### Teoría

1. ¿Bajo que condiciones recomendaría volcar la información de un archivo a un vector?. ¿Qué ventajas obtendría?.
2. Explique para qué sirve cada uno de los parámetros que recibe la función **fread** y qué refleja el valor de retorno de dicha función.
3. ¿Para que sirve la función **fseek**?. ¿En qué tipos de archivo puede utilizarse?.

### Práctica

1. *HindaneDarcc* durante su gobierno en *Kashyyyk* dispuso censar a la población *Wookiee*, para esto ordenó establecer 2 equipos, uno para censar la ciudad *Awrathakka* y otro para censar la ciudad de *Kachirho*. De esta manera, obtuvo 2 archivos correspondientes a la población nativa.

Se sabe que:

- Cada uno de los archivos generados por los censistas contiene registros como se muestra a continuación:

```
typedef struct wookiee{
    char    nombre[50];
    float   altura;
    char    genero;
    bool    apto_combate;
} t_wookiee;
```

- Cada uno de los archivos es de tipo binario de acceso secuencial y está ordenado por nombre.

*HindaneDarcc* necesita que se arme **un solo archivo** binario de acceso secuencial ordenado por nombre con la información de los 2 archivos generados por los censistas.

Adicionalmente, quiere saber cuántos *Wookiees* aptos para combate residen en *Kashyyyk* (esto debe realizarse dentro del mismo proceso de unificación de archivos, no debe volver a abrirse ningún archivo).



2. Adicionalmente la empresa que realizó el censo, le acercó a *HindaneDarcc* un archivo de texto separado por ; con información de los *Wookiees* aptos para combate, ordenado por **[ciudad, punto cardinal y nombre de Wookiee]**, como se muestra a continuación:

```
ciudad;punto_cardinal;nombre
Awraithakka;Este;Attichitcuk
Awraithakka;Este;Chewbacca
...
Awraithakka;Norte;Groznik
Awraithakka;Norte;Hanharr
...
Kachirho;Este;Kirlocca
Kachirho;Este;Lowbacca
Kachirho;Este;Lumpawaroo
...
Kachirho;Sur;Tarfful
Kachirho;Sur;Tyvokka
Kachirho;Sur;Yarua
```

Se quiere generar un listado de los *Wookiees*, agrupados por ciudad y punto cardinal, como se muestra a continuación:

```
Ciudad: Awraithakka
—— Punto Cardinal: Este
—— Nombre: Attichitcuk
—— Nombre: Chewbacca
...
—— Totalizando NNNNN Wookiees en el punto cardinal Este
—— Punto Cardinal: Norte
—— Nombre: Groznik
—— Nombre: Hanharr
...
...
Totalizando MMMM Wookiees en la ciudad de Awraithakka

Ciudad: Kachirho
...
...
Totalizando OOOOO Wookiees en la ciudad de Kachirho

Totalizando PPPPP Wookiees en Kashyyyk
```

3. *Chewbacca* decidió romper el chanchito y contar cuánto *CréditoGalactico* ahorró, para esto, volcó los valores a un vector y sumó el total con el siguiente algoritmo:

```
int sumar_ahorros(int chanchito[], int tope){
    int i = 0;
    int ahorros = 0;
    while(i < tope){
        ahorros = ahorros + chanchito[i];
        i++;
    }
}
```

Se pide crear un algoritmo que resuelva el mismo problema, recursivamente.

4. Todos los *Droides* tienen un método de desbloqueo y *Lumpawaroo* (hijo de *Chewbacca*) lo ha descubierto. Es sabido que solo se ven (*Lumpawaroo* y su padre) para *ElDiaDeLaVida* por lo que quiere compartirse ya que puede ser de mucha ayuda para los *Rebeldes* el poder desarticular a los *Droides* y falta poco para ese día.

Lo que *Lumpawaroo* descubrió es que cada *Droide* trae consigo 3 números, el primero corresponde al **inicio**, el segundo al **incremento** y el tercero al **fin**. Para desbloquear un androide hay que ingresar por su teclado la serie de números comprendidos entre el **inicio** y el **fin**, saltados por el **incremento** establecido.

Ejemplo:

Si un Droide tiene inicio = 24, fin = 87, incremento = 10  
Debe teclearse - 24 34 44 54 64 74 84 - y el mismo quedara desbloqueado.

Se pide crear una rutina recursiva que reciba como parámetro el **inicio**, **fin** e **incremento**, y muestre por pantalla los números a teclear en el *Droide* para desbloquearlo.