

Nombre y Apellido	Nota
-------------------	------

Se tienen las siguientes estructuras:

```
typedef struct
{
    int nro_identificacion;
    int idEspecie;
    char nombreAnimal[30];
    int poblacion; // cantidad de animales
} stAnimal;
```

```
typedef struct {
    int idEspecie,
    int cantEjemplares, // cantidad de animales de esa especie
    char nombreEspecie[30],
}stEspecie;
```

Obtenido	Valor	Inciso
		1. Hacer una función que cargue un arreglo de animales de 30 registros (tantos como quiera el usuario, pero como máximo 30). Se debe guardar en alguna variable la cantidad de registros cargados y retornarlo. Para esto debe validar que la idEspecie que ingrese el usuario exista en el archivo de “especies.dat” .
		2. Hacer una función recursiva que muestre los elementos del arreglo de animales por pantalla.
		3. Hacer una función que pase del archivo a un arreglo de especies aquellas especies que su cantidad de ejemplares supere las 2. Para esto debera calcular la cantidad de ejemplares de cada especie a partir de los elementos del arreglo.
		4. Hacer una función que pase los animales cuya población sea superior a 100, del arreglo a un archivo. El nombre del archivo debe ser “animales.dat” . La función debe recibir como parámetro al arreglo, la cantidad de elementos cargados y el nombre del archivo (cadena de caracteres).
		5. Hacer una función que busque dentro del archivo un registro con un determinado nro_identificacion. En la misma función, mostrar los tres campos por pantalla. La función debe recibir por parámetro el nombre del archivo y el nro_identificacion a buscar.
		6. Calcular recursivamente el promedio de población de todos los animales.
		7. Calcular el porcentaje de los animales con la población mayor al promedio.

Tabla de puntuación:

% hecho	10	20	30	40	50	60	70	75	80	90	100
Nota	1	2	2	3	4	5	6	7	8	9	10