**IMPORTANTE:**

**● Crear un proyecto con su Apellido y Nombre.**

**● Añadir comentarios a su código identificando cada inciso.**

**● Aplicar buenas prácticas al código**

* **tabulación, nombre de variables y funciones -**

**● Si el programa no compila ni muestra los resultados por consola**

**NO SE CORREGIRÁ**

| **typedef struct {**  **char nombre[50];**  **char telefono[50];**  **int DNI; // valores del 1 al 100**  **}stPersona;** | **typedef struct{**  **stPersona dato;**  **struct nodoPersona\* siguiente;**  **}nodoPersona;** |
| --- | --- |

| **typedef struct {**  **int DNI;**  **int tipoContacto; //1-familia 2-trabajo 3-amigos**  **int activo; //1-true 0-false**  **}stContacto ;** | **typedef struct nodoCliente{**  **stContacto dato;**  **struct nodoContacto\* siguiente;**  **struct nodoContacto\* anterior;**  **}nodoContacto;** |
| --- | --- |

| Obtenido | Valor | **Inciso** |
| --- | --- | --- |
|  | 10 | 1) Hacer una función que dado un **DNI** evalúe si se **encuentra** en la **lista** de **personas**. Tiene que ser **recursiva.** |
|  | 10 | 2) Hacer una función para **crear** un **nodoPersona.** Hacer una función para que un usuario **cargue** una **stPersona.** Validar **rango del DNI.** |
|  | 10 | 3) Hacer una función que **muestre** las **personas** de la **lista.** Tiene que ser **recursiva.** |
|  | 15 | 4) Hacer una función que **agregue** un **nodoPersona** al **final** de una **lista** de **personas.** Hacer una función para cargar varias personas. Verificar antes si está repetida en la lista usando el punto 1. |
|  | 15 | 5) Hacer una función para **crear** un **nodoContacto.** Hacer una función para **cargar** un **stContacto.** Validar **tipoContacto** que sea 1-2-3. Validar que el DNI **exista** en la **lista personas simple.** |
|  | 10 | 6) Hacer una función que **agregue** los **contactos** de manera **ordenada por DNI** a la **lista doble** |
|  | 20 | 7)Hacer una función que **muestre** la **lista doble** de manera **recursiva**. Empezar desde **final** y mostrar desde el **último** al primero. |
|  | 10 | 8) Hacer una función que **cuente** los contactos **activos**. |
|  | ------- | 9) Hacer una función main () que invoque a las funciones anteriores y demuestre el correcto funcionamiento del programa. Automatice la mayor cantidad de carga de información |

**Tabla de puntuación:**

| **Obtenido** | 10 | 20 | 30 | 40 | 50 | **60** | 70 | **80** | 90 | 100 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Nota** | 1 | 2 | 3 | 4 | 5 | **6** | 7 | **8** | 9 | 10 |

****

**Nunca mucho costó poco.**