

IMPORTANTE:

- Crear un proyecto con su Apellido y Nombre.
- Añadir comentarios a su código identificando cada inciso.

Estructuras:

<pre>typedef struct { char nombre[20]; char apellido[20]; int dni; int edad; }stEmpleado;</pre>	<pre>typedef struct { stEmpleado dato; struct nodo* siguiente; }nodo;</pre>
---	---

Obtenido	Valor	Inciso
	20	1-A) Crear la funcion crearPersona . 1-B) Crear la funcion crearNodo utilizando la estructura stEmpleado como parametro. 1-D) Crear la funcio mostrarNodo y mostrar el nodo creado en el punto 1-B. 1-C) Crear una funcion cargarEmpleado que le pida los datos de un empleado al usuario y utilizando las funciones crearPersona y crearNodo , crear un nodo con los datos y mostrar ese nodo por pantalla usando la funcion mostrarNodo .
	10	2) Crear una lista con tres nodos. Cada nodo debera guardar una persona diferente. Puede hardcodear los datos de los empleados. Los nuevos nodos deberan insertarse al principio de la lista utilizando la funcion insertarAlPrincipio .
	20	3-A) Cargar un archivo “empleados.bin” con los nodos de la lista. 3-B) Crear una funcion que cargue una lista con los datos de “empleados.bin”.
	10	4) Hacer una función que busque un empleado por apellido en la lista. Si lo encuentra devuelve 1, si no lo encuentra devuelve 0.
	10	5) Hacer una función que elimine el nodo de una lista que coincida con el apellido del empleado pasado por parámetro. La función recibe un puntero doble a la lista . void eliminarNodo (nodoEmpleado* * lista, char apellido[30])
	15	6) Hacer una función que muestre una lista en forma recursiva . Recuerde utilizar modularizacion, no haga todo en una sola funcion.
	10	7) Implementar las funciones de Pila inicPila y apilar utilizando lista simple. Crear una funcion que pase los elementos de una lista a una Pila.
	5	8) Hacer una función main () que invoque a las funciones anteriores y demuestre el correcto funcionamiento del programa.