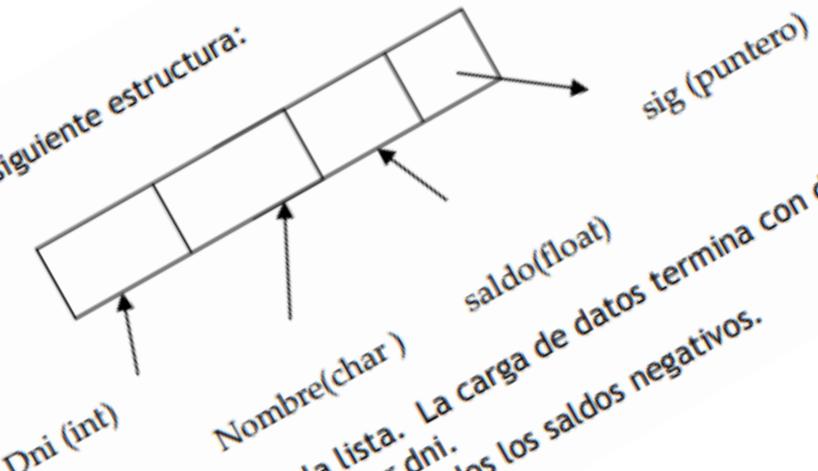
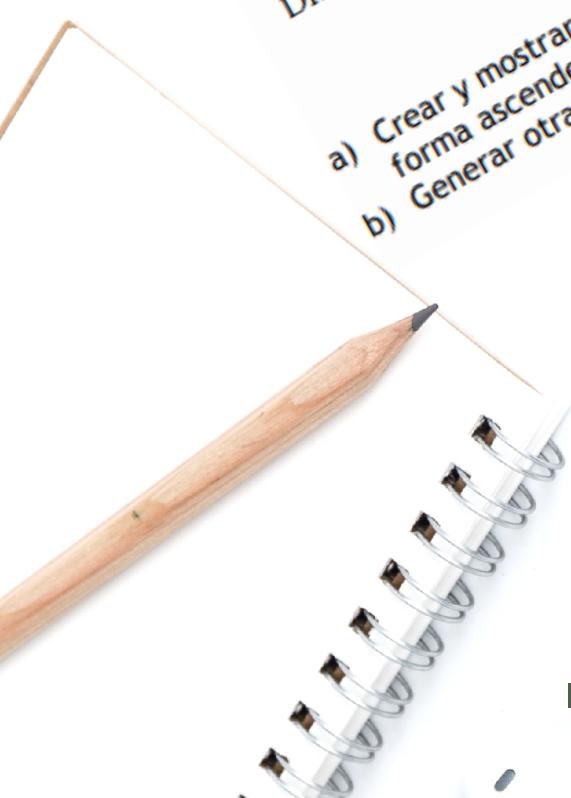


Resolución del ejercicio nº 5

- 5) Dada la siguiente estructura:
- 
- Dni (int) Nombre(char) saldo(float) sig (puntero)
- a) Crear y mostrar la lista. La carga de datos forma ascendente por dni.
b) Generar otra lista con todos los saldos negativos.



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
typedef struct lista{
    int dni;
    char nombre[20];
    float saldo;
    struct lista *sig;
}nodo;
```

Estructura de
la lista

```
void crear (nodo *pt);
void mostrar(nodo *pt);
void generar(nodo *pt, nodo *s);
```

Prototipo de
las funciones

```
int main()
{
    int i=0;
    nodo *prin=NULL;
    nodo *s=NULL;
    prin=(nodo*)malloc(sizeof(nodo));
    s=(nodo*)malloc(sizeof(nodo));
    crear(prin);
    mostrar (prin);
    generar(prin,s);
    printf("la nueva lista con saldos negativos
es \n");
    mostrar (s);
    return 0;
}
```

Reserva de memoria para el primer nodo de la segunda lista y s es el puntero a cabeza de lista.

Reserva de memoria para el primer nodo y prin es el puntero a cabeza de lista

Creamos la lista simplemente enlazada

```
void crear (nodo *registro)
```

```
{
```

```
    printf("Ingrese dni");
```

```
    scanf("%d",&registro->dni);
```

```
    if (registro->dni==0)
```

```
        registro->sig=NULL;
```

```
    else
```

```
{
```

```
    printf("Ingrese nombre");
```

```
    scanf("%s",registro->nombre);
```

```
    printf("Ingrese saldo");
```

```
    scanf("%f",&registro->saldo);
```

```
    registro->sig=(nodo*)malloc(sizeof(nodo));
```

```
    crear (registro->sig);
```

```
}
```

```
return;
```

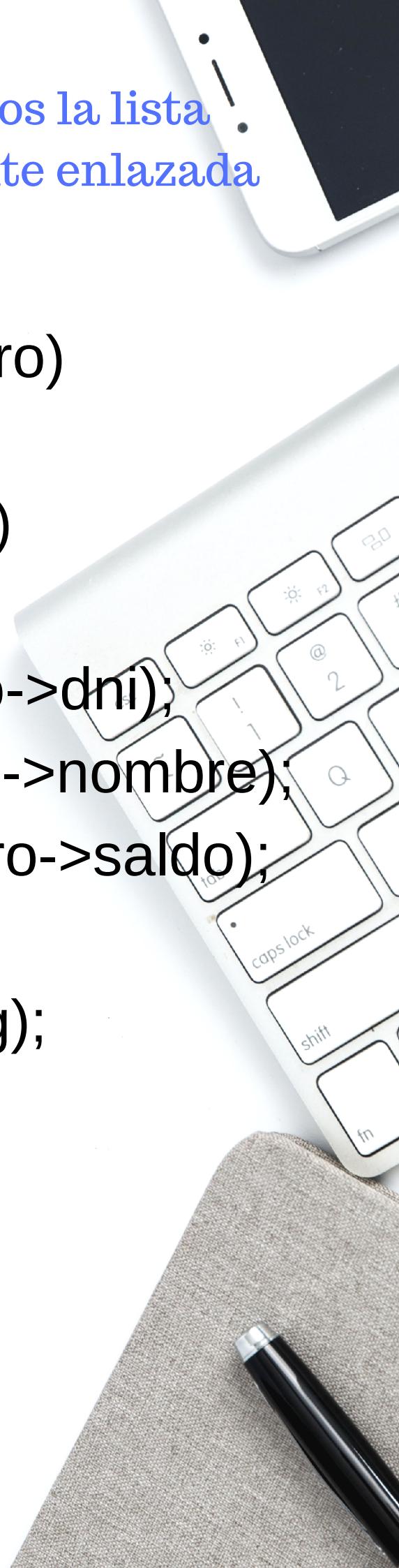
```
}
```

```
v
```

Mostramos la lista
simplemente enlazada

```
void mostrar (nodo *registro)
{
    if (registro->sig !=NULL)
    {
        printf ("%d\n",registro->dni);
        printf ("%s\n",registro->nombre);
        printf ("%,.2f\n",registro->saldo);

        mostrar (registro->sig);
    }
    return;
}
```



```
void generar(nodo *r, nodo *n)
{
    while(r->sig!=NULL)
    {
        if (r->saldo<0)
        {
            n->dni=r->dni;
            strcpy(n->nombre,r->nombre);
            n->saldo=r->saldo;
            n->sig=(nodo*)malloc(sizeof(nodo));
            n=n->sig;
        }
        r=r->sig;
    }
    n->sig=NULL;
}
```

Recorremos la lista hasta llegar a NULL

Pasamos los datos a la lista nueva y reservamos memoria para los próximos datos, avanzando con el puntero

Asignar NULL al último nodo de la lista que generamos