

Se facilitan las siguientes estructuras:

```
struct Tfecha {
    int dia, mes, anio;
};

struct Tcliente {
    char apellidos[40+1];
    struct Tfecha antiguedad;
    float ultimaCompra;
};

struct TlistaClientes{
    int numClientes;
    struct Tcliente arrayClientes[N];
};
```

con las que se define una lista de clientes con los campos descritos en Tcliente. Se debe realizar un programa, implementando para ello los prototipos indicados más abajo, para añadir clientes a la lista, mientras que no se supere la longitud (N) máxima de la lista, y finalmente imprimirlos.

En la siguiente captura de pantalla se muestra un ejemplo de la ejecución:

```
apellidos? lopez_garcia
dia? 1
mes? 1
anio? 2005
ultima compra? 340.35

añadir otro cliente (s/n)?s
apellidos? gutierrez_hernandez
dia? 10
mes? 12
anio? 2018
ultima compra? 120.45

añadir otro cliente (s/n)?n

apellidos: lopez_garcia
antiguedad: 01-01-2005
ultima compra: 340.35

apellidos: gutierrez_hernandez
antiguedad: 10-12-2018
ultima compra: 120.45
```

**SOLUCIÓN:****// includes y defines**

#define N 200

**// estructuras**

```
struct Tfecha {  
    int dia, mes, anio;  
};
```

```
struct Tcliente {  
    char apellidos[40+1];  
    struct Tfecha antiguedad;  
    float ultimaCompra;  
};
```

```
struct TlistaClientes {  
    int numClientes;  
    struct Tcliente arrayClientes[N];  
};
```

**// prototipos**

```
void scanFecha(struct Tfecha *);  
void printFecha (struct Tfecha);  
void scanCliente (struct Tcliente *);  
void printCliente (struct Tcliente);  
void printListaClientes (struct TlistaClientes);  
void inicializarLista (struct TlistaClientes *);  
void anadirCliente (struct TlistaClientes *);  
int longitudLista (struct TlistaClientes);
```

**// main**

```
int main(){  
  
    struct TlistaClientes lista;  
    char eleccion = ' '  
  
    inicializarLista (&lista);  
  
    do{  
        anadirCliente(&lista);  
        system("clear"); //system("clear") == system("cls");  
        printf("\nDesea introducir un cliente mas? (y/n): ");
```

```

        fflush(stdin);
        scanf("%c", &eleccion);
        fflush(stdin);
    } while (eleccion!='y' || eleccion!='Y');

    printf ("\nLa lista tiene %d elementos\n\n", longitudLista (lista));
    printListaClientes (lista);

    return 0;
}

```

#### // implementacion de las funciones

```

//Desarrollo de funciones
void scanFecha(struct Tfecha * fecha){
    printf("Day: ");
    scanf("%d", &(fecha -> day));
    fflush(stdin);
    printf("Month: ");
    scanf("%d", &(fecha -> month));
    printf("Year: ");
    fflush(stdin);
    scanf("%d", &(fecha -> year));
    fflush(stdin);
}

void printFecha (struct Tfecha fecha){
    printf("%d/%d/%d", fecha.day, fecha.month, fecha.year);
}

void scanCliente (struct Tcliente * cliente){
    printf("Surname: ");
    fflush(stdin);
    scanf("%s", cliente -> apellidos);
    fflush(stdin);
    scanFecha(&(cliente -> antiguedad));
    printf("Last purchase :");
    fflush(stdin);
    scanf("%f", &(cliente -> ultimaCompra));
    fflush(stdin);
}

void printCliente (struct Tcliente cliente){
    printf("Surname: %s \n", cliente.apellidos);
    printf("Date: ");
    printFecha(cliente.antiguedad);
}

```

```
    printf("\nLast purchase %.2f", cliente.ultimaCompra);
}

void printListaClientes (struct TlistaClientes listaClientes){
    int i = 0;

    for(i = 0; i < listaClientes.numClientes; i++){
        printCliente(listaClientes.arrayClientes[i]);
        printf("\n\n\n");
    }
}

void inicializarLista (struct TlistaClientes * listaClientes){
    listaClientes -> numClientes = 0;
}

void anadirCliente (struct TlistaClientes * listaClientes){
    scanCliente(&(listaClientes -> arrayClientes[listaClientes ->
numClientes]));
    listaClientes -> numClientes++;
}

int longitudLista (struct TlistaClientes listaClientes){
    return listaClientes.numClientes;
}
```