En esta práctica se pueden usar los **elementos de C permitidos** en prácticas anteriores, y hay que seguir teniendo en cuenta las **restricciones** de funcionamiento de esas prácticas.

Esta práctica está diseñada para continuar ejercitando el uso de **strings**.

Se deben **usar todos y cada uno de los prototipos** indicados en esta práctica, y los de prácticas anteriores que se requieran.

No se utilizará **asignación dinámica de memoria** en ninguna función.

En esta práctica, se deben generar aleatoriamente los DNIs, calculándose la letra, y **ordenar los DNIs ascendentemente,** printando en todo momento lo que se vaya obteniendo. Se usará el algoritmo "bubble sort".

En la comparación de los strings se usará **memcmp()**, no strcmp().

```
 > DNIs aleatorios:
46884664S 59481209N 47402575N 06641707C 87440137W 94254264B 29234255J 76476109M 20739214E 41152496E 48776426G 06865193S
26677393S 48627120Z 62791577J 30292661M 07670409R 49705116T 02776877H 86916169C 21420935T 82868858L 68985614N 72884342X
18568024D 65825414N 64397265W 05190234P 92162008L 63826063M 76886178F 91533683P 14549003P 87434537Z 53776895C 00221112J
27996934T 63844178L 40273476Q 51634812L 60367320W 31354239H 85317836A 41334713X 17054465B 51730417J 56906589N 80165462C
54574124T 58371648V 71812375G 96331012C 28758136V 85838404N 87985855E 60151677F 99532928Z 83105610D 55792176H 22261278S
55331198Y 37905162N 32935862T 74330814N 05581158R 92844508V 49076738M 32403565S 71198771V 67318714Z 33139721X 51160753J
88674388G 73097651V 89768803X 30807015X 90037934H 33859772T 50993718G 81589174D 65953785C 05570013B 61960893C 46553021D
76756906H 35609181Y 24660557A 49911178M 31081934X 24094801R 57080361L 29190313R 23174989M 14838135F 01962262V 02246632S
14484359V 69724430E 42143996S 48829596K

 > DNIs ordenados:
00221112J 01962262V 02246632S 02776877H 05190234P 05570013B 05581158R 06641707C 06865193S 07670409R 14484359V 14549003P
14838135F 17054465B 18568024D 20739214E 21420935T 22261278S 23174989M 24094801R 24660557A 26677393S 27996934T 28758136V
29190313R 29234255J 30292661M 30807015X 31081934X 31354239H 32403565S 32935862T 33139721X 33859772T 35609181Y 37905162N
40273476Q 41152496E 41334713X 42143996S 46553021D 46884664S 47402575N 48627120Z 48776426G 48829596K 49076738M 49705116T
49911178M 50993718G 51160753J 51634812L 51730417J 53776895C 54574124T 55331198Y 55792176H 56906589N 57080361L 58371648V
59481209N 60151677F 60367320W 61960893C 62791577J 63826063M 63844178L 64397265W 65825414N 65953785C 67318714Z 68985614N
69724430E 71198771V 71812375G 72884342X 73097651V 74330814N 76476109M 76756906H 76886178F 80165462C 81589174D 82868858L
83105610D 85317836A 85838404N 86916169C 87434537Z 87440137W 87985855E 88674388G 89768803X 90037934H 91533683P 92162008L
92844508V 94254264B 96331012C 99532928Z
```

Figura 1. Ejemplo de ejecución del programa

```
//includes
#include "stdio.h"
#include "time.h"  // time()
#include "stdlib.h" // srand(), rand()

//defines
#define N 100


//prototipos usados en practicas anteriores
void rand_str_DNI(char str_DNI[9+1]);
void rand_strings_DNIs(char string_DNIs[N][9+1]);
void print_strings_DNIs(char string_DNIs[N][9+1]);

//prototipos de esta práctica
void strings_swap(char [9+1], char [9+1]);
```

```
void strings_bubbleSort(char [N][9+1]);

//main

int main()
{
   srand(time(NULL));
   char string_DNIs[N][9 + 1];

   rand_strings_DNIs(string_DNIs);
   print_strings_DNIs(string_DNIs);
   printf("\nDNIs Ordenados: \n");
   printf("\n");
   strings_bubbleSort(string_DNIs);
   print_strings_DNIs(string_DNIs);


   return 0;
}

// definición de las funciones
void rand_str_DNI(char str_DNI[9+1]){
   int i,Digito,DNI = 0,resto,letraCalculada;
   char letraCalculadaDNI[] = {'T', 'R', 'W', 'A', 'G', 'M', 'Y', 'F', 'P', 'D', 'X',
   'B', 'N', 'J', 'Z', 'S', 'Q', 'V', 'H', 'L', 'C', 'K', 'E'};
   for(i = 0;i < 9;i++){
      Digito = rand() % 10;// numero del 0 - 9
      DNI = DNI * 10 + Digito;
      str_DNI[i] = Digito + '0' ;
   }
   resto = DNI % 23;
   letraCalculada = letraCalculadaDNI[resto];
   str_DNI[8] = letraCalculada;
   str_DNI[9] = '\0';

}

void rand_strings_DNIs(char string_DNIs[N][9+1]){
   int i,j;
   char str_DNI[9+1];
   for(i = 0;i < N;i++){
      rand_str_DNI(str_DNI);
      for(j = 0;j < 10;j++){
         string_DNIs[i][j] = str_DNI[j];
      }
   }
}

void print_strings_DNIs(char string_DNIs[N][9+1]){
   int i;
   for(i = 0;i < N;i++){
      printf("\t%s",string_DNIs[i]);
   }
```

```
    printf("\n");
}

void strings_swap(char StringDNI1[9+1], char StringDNI2[9+1]){
    char aux [9+1];
    memcpy(aux,StringDNI1,9+1);
    memcpy(StringDNI1,StringDNI2,9+1);
    memcpy(StringDNI2,aux,9+1);
}

void strings_bubbleSort(char DNIs[N][9+1]){
    int i,j;
    for (i = N - 1; i >= 0; i--) {
        for (j = 0; j < i; j++) {
            if (memcmp(DNIs[j],DNIs[j+1],9) > 0){
                strings_swap(DNIs[j],DNIs[j+1]);
            }
        }
    }
}
```