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 seguir ejercitando el uso de **strings** (cadenas de caracteres).

Se deben **usar todos los prototipos** indicados y sólo esos prototipos, aparte de alguno que se haya usado en prácticas anteriores.

En la práctica actual, se deben generar aleatoriamente los DNIs, calculándose la letra correspondiente, pero usando strings. Los prototipos de las funciones a implementar son los siguientes:

```
void buscar_substring(char [N][9+1]);
void buscar_letra(char [N][9+1]);
void print_menu ();
void print_strings_DNIs(char [N][9+1]);
void rand_strings_DNIs(char [N][9+1]);
void rand_str_DNI(char [9+1]);
```

Se usará getch() para las opciones del menú. En esta ocasión, no se utilizará asignación dinámica de memoria (malloc, free).

```
55743874Q 71540096E 62854400T 69296374L 28848158V 13326026X 11483652M 31493063Z
43508615K 72244432Y 15209844J 37181739P 97297625N 29696384A 46022741V 55125119F
56308144G 46813789A 81507001S 87172761R 50610073E 94508485J 17344168Y 29041761Y
54408877P 09515274J 31746522J 35679934B 23969142Z 66652743P 33050954T 89758648K
19809854T 40969932D 05219378B 11263151M 91639063W 53588951D 97006296R 79001771Z
14311349J 37040281T 21104891E 16569693X 86948423M 07789334Q 85254888Y 43828050D
30462386J 94448249Z 49465567L 95787978Q 03270350A 87909568A 33707916J 72455517C
47891325N 01478343H 03609734E 66516033X 17607704P 82724985B 26308723N 56289851L
58641081M 32410038W 43609471E 29174516M 50539664Q 42780351Y 47039166A 78393584Q
03026230M 39730012L 96813183L 13441769V 28400751Y 38691591W 06786249F 57767237T
76820528E 63565901L 20057020P 85557317P 22421109L 68903990S 23389320E 53253621L
76202674V 89664776N 32806769Y 04486109M 50654730J 72067278K 25853000B 40584136S
66387249A 45253890X 10375415T 47962971J

1 buscar letra
2 buscar substring
0 END

letra a buscar ? P
 > 37181739P  > 54408877P  > 66652743P  > 17607704P  > 20057020P  > 85557317P

letra a buscar ? N
 > 97297625N  > 47891325N  > 26308723N  > 89664776N

substring a buscar ? 3L
 > 96813183L

substring a buscar ? 00
 > 71540096E  > 62854400T  > 81507001S  > 50610073E  > 97006296R  > 79001771Z  >
 32410038W  > 39730012L  > 28400751Y  > 20057020P  > 25853000B

Process returned 0 (0x0)   execution time : 25.050 s
Press any key to continue.
```

Figura 1. Ejemplo de ejecución del programa

```
//includes

//defines
```

```
//prototipos usados en practicas anteriores
unsigned resto_DNI(unsigned);
char letra_calculada(unsigned );

//prototipos de esta práctica
void buscar_substring(char _string_DNIs[N][9+1]);
void buscar_letra(char _string_DNIs[N][9+1]);
void print_menu ();
void print_strings_DNIs(char string_DNIs[N][9+1]);
void rand_strings_DNIs(char string_DNIs[N][9+1]);
void rand_str_DNI(char str_DNI[9+1]);


//main


int main()
{
char matriz_DNIs[N][10],opcion;
    srand(time(NULL));
    rand_strings_DNIs(matriz_DNIs);
    print_strings_DNIs(matriz_DNIs);
    print_menu();
    do{
    fflush(stdin);
    opcion = getch();
    if(opcion == '1')
        buscar_letra(matriz_DNIs);
    else if(opcion == '2')
        buscar_substring(matriz_DNIs);


    } while (opcion != '0');
    return 0;

}
// definiciones de las funciones
void rand_str_DNI(char DNI[10]){
int i, num;

 for(i=0;i<8;i++)  //Genera la parte numerica random
    DNI[i] = rand()%10 + '0';
for(i=0, num = 0;i<8;i++) //Pasa la string a un int
        num = num*10 + (DNI[i] - '0');
 DNI[i]=letra_calculada(resto_DNI(num));
 DNI[i+1]='\0';
}
```

```
unsigned resto_DNI(unsigned num){
 return (num%23);
}

char letra_calculada(unsigned resto){
char letra[23] =
{'T','R','W','A','G','M','Y','F','P','D','X','B','N','J','Z','S','Q','V','H','L','C','K','E'};
return letra[resto];
}



void rand_strings_DNIs(char matriz_DNIs [N][10]){
int i;
char DNI[10];
for(i=0;i<N;i++){
    rand_str_DNI(DNI);
    strcpy(matriz_DNIs[i],DNI);
  }
}

void print_strings_DNIs(char matriz_DNIs [N][10]){
    int i;
 for(i=0;i<N;i++)
    printf("%s ", matriz_DNIs[i]);
printf("\n");
}

void print_menu(){
 printf("\n1 buscar letra\n");
 printf("2 buscar substring\n");
 printf("0 para acabar(END)\n");
}

void buscar_letra(char matriz_DNIs[N][10]){
    char letra;
    int i;

printf("Letra a buscar?: \n");
do{
   fflush(stdin);
   letra = getch();
   if (letra >= 'a' && letra <= 'z')
        letra-=32;
  } while(letra < 'A' || letra > 'Z');

for(i=0;i<N;i++){
    if(strchr(matriz_DNIs[i], letra)){
        printf("> %s  ", matriz_DNIs[i]);
    }
  }
```

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

void buscar_substring(char matriz_DNIs[N][10]){
int i;
    char substring[10];
printf("Substring a buscar?: \n");
  fflush(stdin);
  scanf("%s", substring);

for(i=0;i<N;i++){
    if(strstr(matriz_DNIs[i], substring)){
        printf("> %s  ", matriz_DNIs[i]);
    }
  }
  printf("\n");
}
```