

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 usando **funciones**, empezar a usar la asignación dinámica de memoria, y los arrays unidimensionales.

De momento, **no está permitido** el uso de strings, u otros elementos del lenguaje C.

En la práctica actual, se deben usar algunas de las funciones definidas en la práctica anterior, definir la función **validar_letra_DNI(unsigned, char);** y usar dicha función invocándola desde el main. En el main, se utilizará **asignación dinámica de memoria** (malloc, free) para todas las variables. La función letra_calculada (unsigned) se definirá declarando un array de chars (sin if ni switch).

```
DNI digito 1 ? 0
DNI digito 2 ? 5
DNI digito 3 ? 6
DNI digito 4 ? 7
DNI digito 5 ? 8
DNI digito 6 ? 9
DNI digito 7 ? 0
DNI digito 8 ? 1
> DNI: 5678901
letra DNI ? A
> resto DNI: 17
> letra DNI introducida incorrecta
> letra correcta: V
```

Figura 1. Ejemplo de ejecución del programa

```
// includes
```

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

```
// prototipos de las funciones
```

```
void scan_dig(char * , unsigned );  
void scan_DNI(unsigned *);  
unsigned resto_DNI(unsigned );  
void scan_letra(char *);  
char letra_calculada(unsigned );  
void validar_letra_DNI(unsigned, char);
```

```
// main
```

```
int main(){
```

```
    char *letrasc = (char *)malloc(sizeof(char));  
    unsigned *resto = (unsigned *)malloc(sizeof(unsigned)), *num = (unsigned *)malloc(sizeof(unsigned));  
    *num = 0;  
    scan_DNI(num);
```

```
        scan_letra(letrasc);  
        fflush(stdin);  
        printf("\n >Letra leida: %c\n", *letrasc);  
        *resto=resto_DNI(*num);  
        printf("\n> RestoDNI: %u\n", *resto);  
        validar_letra_DNI(*resto, *letrasc);  
        free(letrasc);  
        free(resto);  
        free (num);
```

```
}
```

```
// definición de las funciones
```

```
void scan_DNI(unsigned *num){  
    unsigned i,act;  
    char car;  
    for(i=1;i<=8;i++){  
        printf("DNI digito %u: ", i);  
        scan_dig(&car);  
        fflush(stdin);  
        act = car - '0';  
        *num=(*num)*10+act;  
    }  
}
```

```
    printf("\n > DNI: %u", *num);  
}
```

```
void scan_letra(char *letrasc){  
    printf("\n¿Letra?\n");  
    fflush(stdin);  
    scanf(" %c", letrasc);  
    if((*letrasc>='a' && (*letrasc)<='z')  
        (*letrasc)-=32;  
}
```

```
void scan_dig(char *car){  
    do{  
        fflush(stdin);  
        scanf("%c", car);  
    } while(*car>'9' || *car <'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 validar_letra_DNI(unsigned resto, char letrasc){  
    char letra;  
  
    letra=letra_calculada(resto);  
    if(letra==letrasc)  
        printf("> Letra correcta!\n");  
    else {  
        printf("> Letra introducida incorrecta\n");  
        printf("> Letra correcta: %c\n",letra);  
    }  
}
```