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).

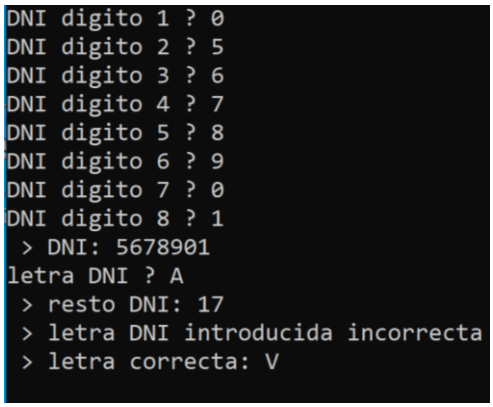

Figura 1. Ejemplo de ejecución del programa

```
// includes
#include "stdio.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 *letraCalculada,*letra;
    unsigned *DNI;

    letraCalculada = (char *)malloc(sizeof(char));
    DNI = (unsigned *)malloc(sizeof(unsigned));
    letra = (char* )malloc(sizeof(char));

    scan_DNI(DNI);
    scan_letra(letra);
    printf("\nRestoDNI: %d",resto_DNI(*DNI));
    *letraCalculada = letra_calculada(*DNI);
    validar_letra_DNI(*letra, *letraCalculada);

    free(DNI);
    free(letraCalculada);
    free(letra);

    return 0;
}

// definición de las funciones

void scan_dig(char *dig, unsigned i) {
    printf("DNI digito %d ? ", i);
    fflush(stdin);
    scanf("%c", dig);
}

void scan_DNI(unsigned *DNI) {
    *DNI = 0;
    char digito;
    for (int i = 1; i <= 8; i++) {
```

```
        scan_dig(&digito, i);
        while (digito < '0' || digito > '9') {
            scan_dig(&digito, i);
        }

        *DNI = *DNI * 10 + (digito - '0');
    }
    printf("\nDNI: %08d       ", *DNI);
}

unsigned resto_DNI(unsigned DNI) {
    return DNI % 23;
}

char letra_calculada(unsigned DNI) {
    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'};
    return letraCalculadaDNI[resto_DNI(DNI)];
}

void scan_letra(char *letra) {
    printf("\nletra DNI ? ");
    fflush(stdin);
    scanf("%c", letra);
    while(*letra < 'A' || *letra > 'Z'){
        scanf("%c", letra);
    }
}

void validar_letra_DNI (unsigned letra, char letra_calculada) {
    if (letra == letra_calculada) {
        printf("\nLetra correcta: %c \n",letra);
        } else {
                printf("\nLetra DNI introducida incorrecta.");
                printf("\nLetra correcta: %c\n",letra_calculada);
    }
}
```