

**En esta práctica** se va a trabajar la programación de cadenas de caracteres y el uso de la función `rand()` para generar valores aleatorios de distinto tipo. Para ello se pide programar las funciones que se describen a continuación y un programa principal que las pruebe.

1. Escribir una función que, pasándole como parámetro un array de N caracteres, almacene en él una cadena caracteres alfanuméricos generados aleatoriamente de la siguiente forma:
  - a. Generar un número aleatorio entre 0 y N-2 que será la longitud de la cadena a generar.
  - b. Para la longitud generada, rellenar la cadena aleatoriamente con mayúsculas, minúsculas y dígitos de la siguiente forma:
    - i. Generar un valor entre 0 y 2, y en función de él generar el carácter:
      1. Si el valor es un 0, generar una mayúscula.
      2. Si el valor es un 1, generar una minúscula.
      3. Si el valor es un 2, generar un dígito.
  - c. Tener en cuenta que al final se deberá añadir el carácter `'\0'`.
2. Escribir una función que, pasándole como parámetro dos cadenas de caracteres, elimine de la primera aquellos caracteres que aparecen en la segunda.
3. Escribir una función que, pasándole como parámetro una cadena de caracteres y un carácter, devuelva la posición, empezando por el final de la cadena, en la que se encuentra el carácter, y un -1 en caso de que no esté.
4. Escribir un programa principal que pruebe las funciones de los apartados anteriores.

Una posible ejecución podría ser:

```
y78p279P22KQLu
IGC3B
```

```
y78p279P22KQLu
IGC3B
```

```
aV3x3iT1rsrse3
ZC96h27Aq8q
```

```
aV3x3iT1rsrse3
El caracter 1 esta en la posición 7
```

```
ZC96h27Aq8q
El caracter 2 esta en la posición 5
```

```
¿Repetir la ejecucion?(s/n):
```

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <string.h>

#define N 25

void rellenar (char *);
void eliminar (char *, char *);
int posChar (char *, char);

int main() {
    char cad1[N];
    char cad2[N];
    char car1, car2, eleccion;

    do{
        srand(time(NULL));
        printf("\tRellenar cadenas\n");
        rellenar (cad1);
        rellenar(cad2);
        printf("Cadena 1:\t%s\nCadena2:\t%s\n", cad1, cad2);
        printf("\n\tEliminar caracter en comun\n");
        eliminar (cad1, cad2);
        printf("Cadena 1:\t%s\n", cad1);
        printf("Cadena2:\t%s\n", cad2);
        printf("\n\tPosicion de un caracter\n");
        printf("Cadena 1:\t%s\n", cad1);
        printf("Escriba un caracter: ");
        fflush(stdin);
        scanf("%c", &car1);
        printf("\nEl caracter %c esta en la poscion %d\n",car1, posChar(cad1, car1));
        printf("\nCadena2:\t%s\n", cad2);
        printf("Escriba un caracter: ");
        fflush(stdin);
        scanf("%c", &car2);
        printf("\nEl caracter %c esta en la poscion %d\n",car2, posChar(cad2, car2));
        printf("Repetir la ejecucion? (s/n): ");
        fflush(stdin);
        scanf("%c", &eleccion);
    } while(eleccion == 's' || eleccion == 'S');

    return 0;
```

```
}

//Definición de funciones
void rellenar(char *cad){
    int longitud = rand() % (N - 1);

    //printf("La longitud es: %d\n", longitud); //Comprobacion
    for(int i = 0; i < longitud; i++){
        int valor = rand() % 3;
        //printf("El valor es: %d\n", valor); //Comprobacion
        if(valor == 0){
            cad[i] = rand() % ('Z' - 'A') + 'A';
        } else if(valor == 1){
            cad[i] = rand() % ('z' - 'a') + 'a';
        } else{
            cad[i] = rand() % ('9' - '0') + '0';
        }
        cad[longitud] = '\0';
    }
}

void eliminar(char *cad1, char *cad2){
    int lengthCad1 = strlen(cad1);
    int lengthCad2 = strlen(cad2);

    for(int i = 0; i < lengthCad1; i++){
        for(int j = i; j < lengthCad2; j++){
            if(cad1[i] == cad2[j]){
                cad2[j] = cad2[j + 1];
            }
        }
        lengthCad1--;
        i--;
    }
}

int posChar(char *cad, char car){
    int pos = -1;

    for(int i = 1; i <= N - 1; i++){
        if(cad[i] == car){
            pos = i;
        }
    }
}
```

```
    return pos;  
}
```