

En esta práctica se pide la implementación de un conjunto de funciones y un programa principal para el manejo de un array de 30 caracteres. En concreto, las funciones a implementar son:

- **rellenar** el array con letras mayúsculas comprendidas entre la A y la F.
- **imprimir** el contenido del array en una línea, separando las letras por un espacio en blanco.
- **eliminar** del array una letra que se pasa como parámetro a la función, cambiándola por el carácter '.' (punto).
- **compactar** el array, de forma que las letras queden a la izda del mismo y los puntos a la dcha.

El programa principal deberá imprimir el contenido del array, al menos, cada vez que sufra algún tipo de modificación. Cuando llame a la función que elimina una letra del array, antes deberá pedir por teclado esta letra y asegurarse de que es una mayúscula entre la A y la F.

Se dispone de la función *int aleatorio (int, int)* que, pasándole como parámetro dos enteros que representan un intervalo, devuelve un valor aleatorio en ese intervalo (esta función se utilizará en prácticas posteriores). Para el correcto funcionamiento de esta función se deberá incluir al principio del programa la librería `time.h`, así como incluir la siguiente instrucción después de la declaración de vbles: `srand(time(NULL));`. A continuación se muestra el esquema del programa con estas indicaciones y la implementación de la función aleatorio, y se tendrá que completar con el código correspondiente a la solución de la práctica.

Una posible ejecución de la misma es la siguiente:

```
A D A C E E B F D F F D B D A E F D C E E B D C C B B B A A

Introduzca una letra mayuscula: 3

Introduzca una letra mayuscula: a

Introduzca una letra mayuscula: R

Introduzca una letra mayuscula: G

Introduzca una letra mayuscula: B

A D A C E E . F D F F D . D A E F D C E E . D C C . . . A A

A D A C E E F D F F D D A E F D C E E D C C A A . . . . .

Process returned 10 (0xA)   execution time : 26.280 s
Press any key to continue.
```

Solución:

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

#define SIZE 30

int aleatorio (int, int);
void imprimir (char *);
void rellenar (char *);
void eliminar (char *, char);
void compactar (char *);

void main ()
{
    char arr[SIZE], c;
    srand(time(NULL));
    rellenar(arr);
    do
    {
        printf("\n\nIntroduzca una letra mayuscula: ");
        scanf("%c", &c);
        fflush(stdin);
    } while (c < 'A' || c > 'F');
    eliminar(arr, c);
    compactar(arr);
}

// ===== FUNCIONES

int aleatorio (int inf, int sup)
{
    int aux = inf;

    if (inf > sup)
    {
        inf = sup;
        sup = aux;
    }
    return (rand()%(sup-inf+1)+inf);
}

void imprimir (char * arr)
{
    printf("\n\n");
    int i;
    for (i = 0; i < SIZE - 1; i++)
        printf("%c ", arr[i]);
    printf("\n");
}
```

```
}

void rellenar(char * arr)
{
    int i;
    for (i = 0; i < SIZE - 1; i++)
        arr[i] = aleatorio('A', 'F');
    imprimir(arr);
}

void eliminar (char * arr, char c)
{
    int i;
    for (i = 0; i < SIZE - 1; i++)
        if (arr[i] == c)
            arr[i] = '.';
    imprimir(arr);
}

void compactar(char * arr)
{
    int i, ult; // ult es el ultimo char a la izquierda
    for (i = ult = 0; i < SIZE - 1; i++)
        /* Comprobamos si el char tiene que ser cambiado de lugar. Para ello, que
        el char actual no sea un blanco y que el ultimo char sea un blanco ('.') */
        if (arr[i] != '.' && arr[ult] == '.')
        {
            arr[ult] = arr[i]; // Movemos el char para la última posición
            arr[i] = '.'; // Hacemos que la posición anterior sea un blanco
            ult = arr[ult+1] == '.' ? ult+1 : i; /* Si el antiguo ultimo blanco + 1 es un blanco, el ultimo char
pasa a ser ult+1
                                     sino, el ultimo pasa a ser el mismo char */
        }
        /* Comprobamos si el char es un blanco, si lo es, y el ultimo char no es un
        blanco, él es el ultimo char */
        else if (arr[i] == '.' && arr[ult] != '.')
            ult = i;
    imprimir(arr);
}
```