

La Primitiva es un sorteo que se celebra dos veces cada semana, consiste en elegir 6 números entre el 1 y el 49 y si coinciden, total o parcialmente, con los elegidos en el sorteo, se obtiene un premio en metálico en función del dinero recaudado para el sorteo.

En esta práctica se pretende simular un sorteo simplificado de La Primitiva. Para ello se realizarán dos programas diferentes:

1. Un **primer programa** que almacene en un fichero binario, cuyo nombre se leerá por teclado, los datos de las apuestas jugadas por los usuarios. Cada apuesta consiste en:
 - 1) Un identificador, que es una cadena de caracteres con el formato “28DDDDDDL”, donde cada D es un dígito y la L una letra del alfabeto. Tanto los dígitos como las letras se generarán aleatoriamente.
 - 2) Un array con los 6 números de la apuesta, que no podrán estar repetidos y que se generarán mediante una función que pasándole este array devuelva en él una combinación válida.

Se deberá definir previamente el tipo de datos APUESTA.

El número total de apuestas a generar y almacenar en el fichero se deberá preguntar al usuario.

2. Un **segundo programa**, distinto al anterior, que:
 - 1) Realice el sorteo, generando los 6 números premiados (usando para ello la función implementada en el programa anterior.
 - 2) Escriba en un fichero de texto, en la primera línea, los números premiados.
 - 3) Lea, una a una, todas las apuestas almacenadas en el fichero generado con el primer programa, y cuyo nombre se deberá introducir por teclado. Por cada apuesta se comprobará si tiene premio o no y, en caso de que sí, se escribirá una línea en el fichero de texto con los datos de la apuesta, el número de aciertos y los números acertados de la apuesta. Para esto **se implementará una función** que, pasándole los números del sorteo, los jugados en la apuesta y un tercer array donde guarde los números acertados, devuelva el número total de aciertos de la apuesta.

Una vez ejecutados los dos programas, en el fichero de texto estarán todas las apuestas premiadas. Se deberá revisar visualmente que los datos son correctos, es decir, que el número de aciertos de cada apuesta son acordes al resultado del sorteo.

Para probar el segundo programa se deberá copiar el fichero generado en el primero en la carpeta en la que está el segundo programa.

Una vez que los dos programas funcionan correctamente se llevarán a cabo las siguientes acciones:

1. **Crear una carpeta nueva en el disco.**
2. **Copiar en esta carpeta los dos programas ejecutables. Estos son del tipo aplicación y están en la carpeta bin/debug de cada proyecto.**
3. **Ir a esta nueva carpeta.**
4. **Ejecutar el primer programa.**
5. **Comprobar que se ha creado el fichero de apuestas.**
6. **Ejecutar el segundo programa.**
7. **Ver el fichero de texto generado.**

Por ejemplo, para una ejecución del primer programa pidiendo la generación de 200000 apuestas, el resultado obtenido en el fichero de aciertos después de una ejecución del segundo programa ha sido:

Combinación ganadora	39 43 5 12 26 14
28100155Y	3 aciertos: 5 12 26
28692635B	3 aciertos: 43 5 12
28520461V	3 aciertos: 43 5 14
28884558J	3 aciertos: 5 26 14
28065287S	3 aciertos: 39 12 26
28109454W	3 aciertos: 5 26 14
28346341I	3 aciertos: 5 12 26
28261131W	3 aciertos: 39 43 26
28932910G	3 aciertos: 39 12 26
28472591E	3 aciertos: 43 5 14
28172570U	3 aciertos: 39 12 14
28339787A	3 aciertos: 39 43 14
28969129M	3 aciertos: 12 26 14
28472667Z	4 aciertos: 39 5 12 14
28725848F	3 aciertos: 5 12 26
28927674K	3 aciertos: 5 12 26
28081250Z	3 aciertos: 43 12 14
28974201I	3 aciertos: 39 43 12
28085620M	3 aciertos: 39 5 14
28951600M	3 aciertos: 43 12 14
28692513V	3 aciertos: 5 12 14
28600560J	3 aciertos: 39 26 14
28688664C	3 aciertos: 39 43 26
28711604Y	3 aciertos: 12 26 14
28036948N	3 aciertos: 39 5 26
28557878T	4 aciertos: 39 12 26 14
28589337X	3 aciertos: 39 5 26
28401648V	3 aciertos: 43 26 14
28782709D	3 aciertos: 39 5 12
28086436L	4 aciertos: 39 43 12 26
28506314W	3 aciertos: 39 5 26
28188418W	3 aciertos: 39 26 14
28796462H	3 aciertos: 39 5 26

Hay que tener en cuenta que con el mismo fichero de apuestas, por cada ejecución del segundo programa el resultado es diferente, ya que cambian los números elegidos en el sorteo.

SOLUCIÓN PROGRAMA 1:

```
// includes, defines
```

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

```
// structs
```

```
typedef struct{
    char id[11];
    unsigned n[6];
} APUESTA;
```

```
// prototipos
```

```
int randomNum(int, int);
void crearID(char [11]);
void crearNum(int [6]);
void crearApuesta(APUESTA*);
void crearFapuestas(const char *, unsigned);
```

```
// main
```

```
int main(void)
{
    char fichName[20];
    int bets;

    srand(time(NULL));
    printf("\nDonde quieres guardar las apuestas?:");
    gets(fichName);

    printf("\nCuantas apuestas quieres generar?:");
    scanf("%d", &bets);
    fflush(stdin);

    printf("\nEl tamaño del archivo va a ser %.9fmb Seguir? (S/N): ",
(sizeof(APUESTA)/1000000.f));
    if (toupper(getchar()) != 'S'){
        return 1;
    }

    crearFapuestas(fichName, bets);
    return 0;
}
```

```
// implementacion funciones
```

```
int randomNum(int num1, int num2){
    return (rand() % (num2 - num1 + 1) + num1);
}

void crearID(char id[11]){
    int i = 0;
    int j = 0;

    id[0] = '2';
    id[1] = '8';
    for(i = 2; i < 10; i++){
        id[i] = randomNum('0', '9');
    }
    id[10] = randomNum('A', 'Z');
    id[11] = '\\0';
}

void crearNum(int num[6]){
    int i = 0;

    for (i = 0; i < 6; ){
        num[i] = randomNum(1, 49);
        for (j = 0; j < i && num[j] != num[i]; j++);
        if (j == i){
            i++;
        }
    }
}

void crearApuesta(APUESTA* ap){
    crearID(ap -> id);
    crearNum(ap -> n);
}

void crearFapuestas(const char * fichName, unsigned ap){
    int i;
    FILE* fichIn;

    if(!(fichIn = fopen(fichName, "wb"))){
        printf("Error en la apertura de fichero");
        return 0;
    }
    for(i = 0; i < ap; i++){
        crear_apuesta(&ap);
        fwrite(&ap, sizeof(APUESTA), 1, fichIn);
    }
    fclose(fichIn);
}
```

SOLUCIÓN PROGRAMA 2:

```
// includes, defines
```

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

```
// structs
```

```
typedef struct{
    char id[11];
    unsigned n[6];
}APUESTA;
```

```
// prototipos
```

```
int randomNum(int, int);
void crearNum(unsigned[6]);
void premiar(int[6], const char*);
unsigned acertados(APUESTA, unsigned[6], unsigned[]);
```

```
// main
```

```
int main(){
    APUESTA bets;
    FILE* fichBets;
    FILE* fichWinners;
    unsigned premiado[6];
    unsigned acc[6];
    char fichBetsName[20];
    char fichWinnersName[20];
    int i = 0;
    int j = 0;

    srand(time(NULL));

    printf("\nDe donde quieres leer las apuestas? :");
    gets(fichBetsName);

    printf("\nDonde quieres guardar los premiados? :");
    gets(fichWinnersName);

    if(!(fichBets = fopen(IN, "rb")))
        exit(1);

    if(!(fichWinners = fopen(OUT, "w")))
        exit(1);
```

```

    crear_num(premiado);

    fprintf(fichWinners, "La combinacion ganadora es: ");
    for (i=0; i<6; i++){
        fprintf(fichWinners, "%u ", premiado[i]);
    }
    fprintf(fichWinners, "\n");

    while (fread(&bets, sizeof(APUESTA), 1, fichBets)){
        if ((j = acertados(bets, premiado, acc)) > 1){
            fprintf(fichWinners, "%s %u: ", ap.id, j);

            for (i = 0; i<6; i++){
                fprintf(fichWinners, "%2u ", ap.n[i]);
            }
            fprintf(fichWinners, "| ");
            for (i = 0; i<j; i++){
                fprintf(fichWinners, "%2u ", acc[i]);
            }
            fprintf(fichWinners, "\n");
        }
    }

    fclose(fichWinners);
    fclose(fichBets);
    return 0;
}

```

// implementacion funciones

```

int randomNum(int num1, int num2){
    return (rand() % (num2 - num1 + 1) + num1);
}

void crear_num(unsigned num[6]){
    int i = 0;
    int j = 0;

    for (i = 0; i < 6; )
    {
        num[i] = randomNum(1, 49);
        for (j = 0; j < i && num[j] != num[i]; j++);
        if (j == i){
            i++;
        }
    }
}

unsigned acertados(APUESTA bets, unsigned premiado[6], unsigned * acertados)
{

```

```
unsigned aciertos = 0;
int i = 0;
int j = 0;

for (i = 0; i < 6; i++){
    for(j = 0; j < 6 && bets.n[i] != premiado[j]; j++){
        if (j < 6){
            acertados[aciertos++] = bets.n[i];
        }
    }
}
return aciertos;
}
```