

## INSO - Introducción a la programación I – Curso 2024 - 2025

### Ejercicio Evaluable 1

La práctica 1 estará formada por los siguientes 5 ejercicios:

- A. Clave de tres caracteres
- B. Calificaciones alumnos
- C. Entero de tres cifras
- D. Operaciones aritméticas, lógicas y relacionales
- E. Descomposición de un Número Entero de 32 Bits en Partes de 8 Bits

A continuación, encontrarás los detalles de cada enunciado para escribir el código correspondiente. En todos ellos debes realizar el análisis del problema que se plantea, estudiando los datos de entrada que necesitas, plantea a continuación las sentencias de ejecución en base a las operaciones que debes realizar y finalmente muestra el resultado.

Se valorará si se ejecuta correctamente, así como la estructura general, la claridad y la legibilidad. Recuerda siempre utilizar identificadores que sean adecuados y el uso de comentarios. El uso de bucles no es necesario.

## Ejercicio A. Clave de tres caracteres

El programa deberá pedir una clave de 3 caracteres separados por guiones al usuario.

La clave deberá contener una letra mayúscula, una letra minúscula y un dígito, escritos en cualquier orden.

El programa deberá comprobar si la clave es válida o no y sacar el mensaje correspondiente de "Clave correcta" o "Clave incorrecta".

### Ejemplo de ejecución:

```
$/clave.exe
Introduce la clave: u-6-F
La clave es correcta.
$/clave.exe
Introduce la clave: 7-6-F
La clave es incorrecta.
$/clave.exe
Introduce la clave: u6F
La clave es incorrecta.
```

## Ejercicio B- Calificaciones alumnos

Realiza un programa que determine si un estudiante aprueba o suspende una asignatura en función de sus calificaciones en tres pruebas, su examen y su asistencia a clase. El programa debe solicitar:

- Nota de primera prueba (0-10)
- Nota de segunda prueba (0-10)
- Nota de tercera prueba (0-10)
- Nota examen final (0-10)
- Asistencia a clase (0%-100%)

Cálculo de la nota final:

- Pruebas (40%):
  - Prueba 1 (20%)
  - Prueba 2 y 3 (40%) cada una
- Examen (55%)
- Asistencia (5%)

Criterios para aprobar:

- Nota de examen superior o igual a 4, media de las pruebas superior o igual a 5, nota final superior o igual a 5 y asistencia mayor al 80%.
- En caso de suspender algún apartado se debe indicar al usuario utilizando la consola.

**Ejemplo de ejecución:**

```
$/notas.exe
```

```
Nota de la primera prueba: 4.5
```

```
Nota de la segunda prueba: 4.7
```

```
Nota de la tercera prueba: 5.0
```

```
Nota del examen final: 5.0
```

```
Nota de asistencia: 0.9
```

```
El alumno ha suspendido
```

```
Las pruebas no han superado la nota minima.
```

```
$/grades.exe
```

```
Nota de la primera prueba: 7.0
```

```
Nota de la segunda prueba: 6.7
```

```
Nota de la tercera prueba: 5.0
```

```
Nota del examen final: 5.0
```

```
Nota de asistencia: 0.9
```

```
El alumno ha aprobado.
```

### Ejercicio C. Entero de tres cifras

Desarrolla un programa que pida al usuario un número entero positivo de tres cifras. El programa debe clasificar el número en función de sus dígitos:

- Si el número tiene tres dígitos diferentes, debe mostrar "El número tiene tres dígitos únicos".
- Si el número tiene dos dígitos iguales, debe mostrar "El número tiene dos dígitos iguales".
- Si los tres dígitos son iguales, debe mostrar "El número tiene tres dígitos iguales".

#### Ejemplo de ejecución:

```
$/digits.exe
Introduce un numero entero positivo de 3 digitos: 100
El numero tiene dos digitos iguales.
$/digits.exe
Introduce un numero entero positivo de 3 digitos: 123
El numero no tiene ningun digito repetido.
```

### Ejercicio D Operaciones aritméticas, lógicas y relacionales

Escriba un programa en C que solicite al usuario que ingrese dos números enteros. Luego, el programa debe usar operadores relacionales y lógicos para realizar lo siguiente:

1. Verificar e imprimir si el primer número es mayor o igual que el segundo número.
2. Verificar e imprimir si el primer número es menor que 100 y el segundo número es mayor que 50.
3. Verificar e imprimir si alguno de los números es igual a 0 o si ambos números son pares.
4. Verificar e imprimir si los dos números no son iguales.
5. Verificar e imprimir si los dos números son pares.

#### Ejemplo de ejecución:

```
Entrada:
Introduce el primer número: 75
Introduce el segundo número: 60
Salida:
El primer número es mayor o igual que el segundo número: Sí
El primer número es menor que 100 y el segundo número es mayor
que 50: Sí
Uno de los números es 0 o ambos son pares: No
Los dos números son diferentes: Sí
Los dos números son pares: No
```

## Ejercicio E Descomposición de un Número Entero de 32 Bits en Partes de 8 Bits

Escribe un programa en C que solicite al usuario un número entero de 32 bits y luego imprima las cuatro partes de 8 bits de ese número. El programa debe utilizar desplazamientos de bits y operaciones AND para extraer y mostrar cada parte de 8 bits por separado. Usa la función `printBinary` proporcionada para imprimir cada parte en binario.

### Requisitos:

1. Solicitar al usuario que ingrese un número entero de 32 bits.
2. Descomponer el número en cuatro partes de 8 bits utilizando desplazamientos de bits y operaciones AND.
3. Usar la función `printBinary` para imprimir cada parte en binario.

### Función Proporcionada:

```
/*
Asignar 8 al parámetro positions, si se desean mostrar los 8
bits menos significativos (primeros). Asignar 32 si se desea ver
el número de 32 bits completo.
*/
void printBinary(int number, int positions) {
    // Assuming a 32-bit integer
    unsigned int mask = 1 << (positions - 1); // Start with the
leftmost bit
    int i = 0;
    while(i < positions){
        // Print '1' or '0' based on the current bit
        if (number & mask) {
            printf("1");
        } else {
            printf("0");
        }
        // Shift the mask to the right by one bit
        mask >>= 1;
        i++;
    }
    printf("\n");
}
```

### Ejemplo de Ejecución:

```
Ingresa un número entero de 32 bits: 305419896
Parte 1 (bits 31-24): 00010010
Parte 2 (bits 23-16): 00101100
Parte 3 (bits 15-8): 01001101
Parte 4 (bits 7-0): 01101000
```

