

## 1. DATOS INFORMATIVOS

Carrera: Electronica y automatisacion

Asignatura: Fundamentos de programacion

Tema del taller:Correccion de la conjunta

Integrantes: Richard Casa

Fecha: 23/01/2026 Paralelo: 29583

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

#define MAX_INTENTOS 10
#define TAM_TEXTO 30

typedef struct {
    int valor;
    char resultado[TAM_TEXTO]; // "Correcto", "Menor que el secreto", "Mayor que el secreto"
} Intento;

// --- Prototipos ---
int leerEntero(const char *mensaje);
int leerEnteroEnRango(const char *mensaje, int min, int max);
```

```
int esDuplicado(const Intento intentos[], int usados, int valor);

void imprimirResumen(int min, int max, int secreto, const Intento intentos[], int usados, int
maxIntentos);

int main(void) {

    int min, max;

    int maxIntentos;

    int numeroSecreto;

    Intento intentos[MAX_INTENTOS];

    int usados = 0;

    // ----

    // RF01: Configuración del rango

    // ----

    printf("==> CONFIGURACION DEL RANGO ==>\n");

    min = leerEntero("Ingrese el minimo del rango: ");

    max = leerEntero("Ingrese el maximo del rango: ");

    if (min >= max) {

        printf("ERROR: El minimo debe ser menor que el maximo.\n");

        return 1;

    }

    // Inicializar RNG y secreto

    srand((unsigned)time(NULL));
```

```
numeroSecreto = (rand() % (max - min + 1)) + min;
```

```
// -----
```

```
// RF02: Configuración de intentos
```

```
// -----
```

```
printf("\n==> CONFIGURACION DE INTENTOS ==>\n");
```

```
maxIntentos = leerEnteroEnRango("Seleccione el maximo de intentos (1 a 10): ", 1,  
MAX_INTENTOS);
```

```
printf("\n==> COMIENZA EL JUEGO ==>\n");
```

```
// -----
```

```
// Ciclo principal (RF03, RF04)
```

```
// -----
```

```
while (usados < maxIntentos) {
```

```
    int valor;
```

```
    printf("\nIntento %d de %d. Ingrese un numero: ", usados + 1, maxIntentos);
```

```
    if (scanf("%d", &valor) != 1) {
```

```
        // Limpiar buffer en caso de entrada no numérica
```

```
        int c; while ((c = getchar()) != '\n' && c != EOF) {}
```

```
        printf("ERROR: Entrada no valida. Intente nuevamente.\n");
```

```
        continue;
```

```
}
```

```
// Validar rango (RF03)
```

```
if (valor < min || valor > max) {

    printf("ERROR: Numero fuera del rango [%d - %d]\n", min, max);

    continue;

}

// Validar duplicado (RF03)

if (esDuplicado(intentos, usados, valor)) {

    printf("ERROR: Ya ingresaste este numero antes.\n");

    continue;

}

// Registrar intento (RF04)

intentos[usados].valor = valor;

// Calcular resultado textual para RF05

if (valor == numeroSecreto) {

    strncpy(intentos[usados].resultado, "Correcto", TAM_TEXTO - 1);

    intentos[usados].resultado[TAM_TEXTO - 1] = '\0';

} else if (valor < numeroSecreto) {

    strncpy(intentos[usados].resultado, "Menor que el secreto", TAM_TEXTO - 1);

    intentos[usados].resultado[TAM_TEXTO - 1] = '\0';

} else {

    strncpy(intentos[usados].resultado, "Mayor que el secreto", TAM_TEXTO - 1);

    intentos[usados].resultado[TAM_TEXTO - 1] = '\0';

}
```

```
// Mensaje interactivo

if (valor == numeroSecreto) {

    printf("¡FELICIDADES! Adivinaste el numero secreto.\n");

    usados++; // se cuenta el intento ganador

    break;

} else if (valor < numeroSecreto) {

    printf("El numero secreto es MAYOR.\n");

} else {

    printf("El numero secreto es MENOR.\n");

}

usados++;

}

// -----
// RF05: Resumen detallado en tabla
// -----
imprimirResumen(min, max, numeroSecreto, intentos, usados, maxIntentos);

if (usados == maxIntentos && (usados == 0 || intentos[usados - 1].valor != numeroSecreto)) {

    printf("\nNo lograste adivinar el numero. ¡Sigue intentando!\n");

}

return 0;
```

}

// -----

// Implementaciones auxiliares

// -----

```
int leerEntero(const char *mensaje) {
```

```
    int v;
```

```
    for (;;) {
```

```
        printf("%s", mensaje);
```

```
        if (scanf("%d", &v) == 1) {
```

```
            return v;
```

```
        }
```

```
        // limpiar buffer
```

```
        int c; while ((c = getchar()) != '\n' && c != EOF) {}
```

```
        printf("Entrada no valida. Intente nuevamente.\n");
```

```
    }
```

```
}
```

```
int leerEnteroEnRango(const char *mensaje, int min, int max) {
```

```
    int v;
```

```
    for (;;) {
```

```
        printf("%s", mensaje);
```

```
        if (scanf("%d", &v) == 1 && v >= min && v <= max) {
```

```
            return v;
```

}

```
int c; while ((c = getchar()) != '\n' && c != EOF) {}

printf("ERROR: Debe estar entre %d y %d.\n", min, max);

}
```

```
int esDuplicado(const Intento intentos[], int usados, int valor) {
```

```
    for (int i = 0; i < usados; ++i) {

        if (intentos[i].valor == valor) {

            return 1;
        }
    }

    return 0;
}
```

```
void imprimirResumen(int min, int max, int secreto, const Intento intentos[], int usados, int maxIntentos) {
```

```
    printf("\n=====\\n");
    printf(" RESUMEN DETALLADO DEL JUEGO\\n");
    printf("=====\\n");
    printf("Rango elegido: [%d - %d]\\n", min, max);
    printf("Numero secreto: %d\\n", secreto);
    printf("Intentos permitidos: %d\\n", maxIntentos);
    printf("Intentos realizados: %d\\n", usados);
    printf("Intentos restantes: %d\\n\\n", maxIntentos - usados);
```

```
// Tabla
printf("-----\n");
printf("%-10s %-15s %-20s\n", "Intento", "Valor", "Resultado");
printf("-----\n");
for (int i = 0; i < usados; ++i) {
    printf("%-10d %-15d %-20s\n", i + 1, intentos[i].valor, intentos[i].resultado);
}
printf("-----\n");
```

<https://onlinedb.com/MoHc24qDm>

<b>ESPE</b>		DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN																		
EXAMEN DE: FUNDAMENTOS DE PROGRAMACIÓN																				
FORMA A PRÁCTICA																				
PERÍODO: NOMBRE: CARRERA:	PARCIAL: <b>3</b> CURSO (NRC): 704853 FECHA: 21/01/2015	<i>(Handwritten signature)</i>																		
<b>INDICACIONES GENERALES</b> <ul style="list-style-type: none"> <li>• La evaluación es personal y no se permite intercambiar información con sus compañeros, si así sucede la atención por una vez, se pondrá a penas CERO y no recibirá la evaluación.</li> <li>• La parte práctica tiene sobre (En la caja de QHús y formato indicado por el Sr. Docente (Ayudante).</li> <li>• Nombres, UZ_Examen,</li> <li>• Duración de 2h.</li> </ul>																				
<b>Parte Práctica (20 Puntos)</b> <b>Instrucciones:</b> <ol style="list-style-type: none"> <li>1. A continuación, se presentan los requisitos funcionales (RF) para el desarrollo de su programa.</li> <li>2. Recuerde que en el código elaborado por UU. Debe comentar donde desarrolla el RF solicitado, en este documento coloque el código y al final las capturas de pantalla que ilustran cada RF.</li> <li>3. Deje ya estos en su evaluación.</li> <li>4. Este documento debe ser subido en formato PDF con Apellidos Nombres_UZ_Examen, a la tarea el ALUMNADO VIRTUAL.</li> </ol>																				
<b>1. REQUISITOS FUNCIONALES</b> <p>RF01 - Configuración dinámica del rango: El programa debe permitir que el usuario defina el rango, número y máscara del número secreto.</p> <p>RF02 - Configuración dinámica de intentos: El programa debe permitir que el usuario seleccione el número máximo de intentos (hasta 10).</p> <p>RF03 - Validación de rango y duplicados: El programa debe validar que cada intento esté dentro del rango y no se repita.</p> <p>RF04 - Registro de intentos en matriz: El programa debe almacenar en una matriz el número de intento, el valor ingresado y el resultado (0 bajo, 1 alto, 2 correcto).</p> <p>RF05 - Visualización de resumen detallado. Al terminar, el programa debe mostrar una tabla con cada intento y el resultado textual, y revelar el número secreto si no se advinó.</p>																				
<b>2. RUBRICA DE CALIFICACIÓN</b> <table border="1"> <thead> <tr> <th>Criterio</th> <th>4 puntos – Excelente</th> <th>3 puntos – Bueno</th> <th>2 puntos – Aceptable</th> <th>1 punto – Deficiente</th> <th>EVALUACIÓN</th> </tr> </thead> <tbody> <tr> <td>Inicialización y generación del número secreto</td> <td>Número aleatorio generado correctamente y una sola vez.</td> <td>Se genera bien, pero fuera de lugar lógico.</td> <td>Se generan varios números o el rango es incorrecto.</td> <td>No se genera el número correctamente.</td> <td><b>3</b></td> </tr> <tr> <td>Ingreso y validación de datos</td> <td>Captura todos los intentos y valida correctamente.</td> <td>Captura intentos, pero tiene validaciones limitadas.</td> <td>Captura intentos, pero con errores menores.</td> <td>No captura correctamente los intentos.</td> <td><b>3</b></td> </tr> </tbody> </table>			Criterio	4 puntos – Excelente	3 puntos – Bueno	2 puntos – Aceptable	1 punto – Deficiente	EVALUACIÓN	Inicialización y generación del número secreto	Número aleatorio generado correctamente y una sola vez.	Se genera bien, pero fuera de lugar lógico.	Se generan varios números o el rango es incorrecto.	No se genera el número correctamente.	<b>3</b>	Ingreso y validación de datos	Captura todos los intentos y valida correctamente.	Captura intentos, pero tiene validaciones limitadas.	Captura intentos, pero con errores menores.	No captura correctamente los intentos.	<b>3</b>
Criterio	4 puntos – Excelente	3 puntos – Bueno	2 puntos – Aceptable	1 punto – Deficiente	EVALUACIÓN															
Inicialización y generación del número secreto	Número aleatorio generado correctamente y una sola vez.	Se genera bien, pero fuera de lugar lógico.	Se generan varios números o el rango es incorrecto.	No se genera el número correctamente.	<b>3</b>															
Ingreso y validación de datos	Captura todos los intentos y valida correctamente.	Captura intentos, pero tiene validaciones limitadas.	Captura intentos, pero con errores menores.	No captura correctamente los intentos.	<b>3</b>															

Uso de la matriz para organizar datos.	Metrica implementada correctamente y se imprime al final.	Metrica implementada, pero con errores menores.	Uso parcial de la metrica.	No se usa la metrica adecuadamente.	4
Comparación y control de flujos.	Comparación exacta, con mensajes adecuados y flujo lógico.	Mensajes adecuados con mínimo error de flujo.	Comparación basico con mensajes genéricos.	Lógica confusa o incorrecta.	4
Mensajes finales y condición salida.	Mensaje secreto válido solo si acierta, resumen completo.	Muestra el resumen, pero con fallos en el mensaje.	Muestra el mensaje incluso cuando no acierta.	No se muestran resultados correctamente.	4
TOTAL SOBRE 20 PUNTOS					

Elaborado por: Ing. Jenny A Ruiz R.  
Documento TC DCCO  
Fecha: 21/01/2025



**ESPE**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA**

Piano Universitat de les Illes Balears	21	2018	<input checked="" type="checkbox"/>
Alumn Richard Lava			
Curs 2018-19			
Professor			