

1. DATOS INFORMATIVOS

Carrera: Electronica y automatizacion

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://onlinegdb.com/MoHc24qDm>



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

ESPE DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
EXAMEN DE FUNDAMENTOS DE PROGRAMACIÓN
FORMA A PRÁCTICA

PERIODO: Segunda Onda PARCIAL: 3
NOMBRE: Estefanía y Sebastián CURSO (NR): 745.5.3
CARRERA: Administración y Gestión FECHA: 21/01/2022

INDICACIONES GENERALES

- La evaluación es personal y no se permite intercambiar información con los compañeros, si se detecta la atención por una vez, se procederá a ponerle CERO y no tendrá su evaluación.
- La parte práctica debe subirse en la carpeta de GitHub y formata indicada por el Sr. Docente (Apellidos Nombres, UZ, Examen).
- Duración de 2h.

Parte Práctica (20 Puntos)
Instrucciones:

- A continuación, se presentan los requisitos funcionales (RF) para el desarrollo de su programa.
- Recuerde que en el código elaborado por UZ. Debe comentar donde desarrolla el RF solicitado, en este documento coloque el código y al final las capturas de pantalla que evidencien los RF.
- Desde ya éxito en su evaluación.
- Este documento debe ser subido en formato PDF con Apellidos Nombres, UZ, Examen, a la tarro al ALA VIRTUAL.

1. REQUISITOS FUNCIONALES

RF01 - Configuración dinámica del rango. El programa debe permitir que el usuario defina el rango mínimo y máximo del número secreto.

RF02 - Configuración dinámica de intentos. El programa debe permitir que el usuario seleccione el número máximo de intentos (hasta 10).

RF03 - Validación de rango y duplicados. El programa debe validar que cada intento esté dentro del rango y no se repita.

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).

RF05 - Visualización de resumen detallado. Al finalizar, el programa debe mostrar una tabla con cada intento y el resultado textual, y revelar el número secreto si no se adivinó.

2. RUBRICA DE CALIFICACIÓN

Criterio	4 puntos - Excelente	3 puntos - Buena	2 puntos - Aceptable	1 punto - Deficiente	EVALUACIÓN
Inicialización y generación del número aleatorio	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.	3
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.	3



Uso de la matriz para organizar datos.	Matriz implementada correctamente y se incluye el final.	Matriz implementada, pero con errores menores.	Uso parcial de la matriz.	No se usa la matriz adecuadamente.	4
Condiciones y control de flujo.	Comparación exacta, con mensajes adecuados y flujo lógico.	Mensajes adecuados con mínimo error de flujo.	Comparación básica con mensajes genéricos.	Lógica confusa o incorrecta.	4
Mensajes finales y condición secreta.	Mensaje secreto visible 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.
Docente TC DCCO
Fecha: 21/12/2025



ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

Plantel Universitario de las Fuerzas Armadas
Alumno: Richard Irua
Curso: 23553
Profesor:
Asignatura: Fundamentos de programación

Actividad	Fecha	Calificación
1. Introducción a la programación	21/01/2023	8.5
2. Estructuras de datos	22/01/2023	7.5
3. Algoritmos de búsqueda	23/01/2023	6.5
4. Algoritmos de ordenamiento	24/01/2023	7.0
5. Programación orientada a objetos	25/01/2023	8.0
6. Programación de redes	26/01/2023	7.5
7. Programación de bases de datos	27/01/2023	8.5
8. Programación de sistemas de información	28/01/2023	7.5
9. Programación de sistemas de control	29/01/2023	8.0
10. Programación de sistemas de simulación	30/01/2023	7.5

Nota: 7.5

Observaciones: El alumno ha demostrado un buen nivel de comprensión y aplicación de los conceptos de programación. Se recomienda continuar con la práctica de los algoritmos de ordenamiento y programación orientada a objetos.

Revisado por: [Firma]