

Corrección

Nombre: Ederson Gualoto

NRC: 29583

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

int main() {
    int min, max, intentosMax;// rango y máximo de intentos
    int secreto;// número secreto generado por la PC
    int intentos = 0;// contador de intentos válidos realizados
    int encontrado = 0;; // 1 si se adivinó

    // RF01 Configuración dinámica del rango

    printf("Ingrese el valor mínimo del rango: ");
    scanf("%d", &min);

    printf("Ingrese el valor máximo del rango: ");
    scanf("%d", &max);

    // RF02 Configuración dinámica de intentos 10

    do {
        printf("Ingrese el número máximo de intentos (1 a 10): ");
        scanf("%d", &intentosMax);
    } while (intentosMax < 1 || intentosMax > 10);

    // Genero el número secreto
    srand(time(NULL));// siembra el generador con el tiempo actual
    secreto = rand() % (max - min + 1) + min;

    // RF04 Matriz para registrar intentos
    // columnas: [0]=nro intento, [1]=valor ingresado, [2]=resultado
```

```
int registro[10][3];

while (intentos < intentosMax && !encontrado) {

    int numero;

    int repetido = 0;

    printf("\nIntento %d - Ingrese un numero: ", intentos + 1);
    scanf("%d", &numero);

    // RF03 Validar rango
    if (numero < min || numero > max) {
        printf("Numero fuera del rango permitido.\n");
        continue;
    }

    // RF03 Validar duplicados
    for (int i = 0; i < intentos; i++) {
        if (registro[i][1] == numero) {
            repetido = 1;
            break;
        }
    }

    if (repetido) {
        printf("Numero repetido intente de nuevo.\n");
        continue;
    }

    // Registrar intento
    registro[intentos][0] = intentos + 1;// número de intento humano legible
    registro[intentos][1] = numero;

    if (numero < secreto) {
        registro[intentos][2] = 0; // bajo
```

```
printf(" El numero es bajo.\n");
} else if (numero > secreto) {
    registro[intentos][2] = 1; // alto
    printf("El numero es alto.\n");
} else {
    registro[intentos][2] = 2; // correcto
    printf(" Adivinaste el numero.\n");
    encontrado = 1;
}
```

```
intentos++;
}
```

```
// RF05 final
printf("\n== Resumen ==\n");
printf("Intentos | Numero | Resultado\n");
printf("-----\n");
```

```
for (int i = 0; i < intentos; i++) {
    printf(" %d | %d | ",
           registro[i][0], registro[i][1]);
```

```
    if (registro[i][2] == 0)
        printf("Bajo\n");
    else if (registro[i][2] == 1)
        printf("Alto\n");
    else
        printf("Correcto\n");
}
```

```
if (!encontrado) {
    printf(" No se adivino el numero secreto.\n");
    printf("El numero secreto era: %d\n", secreto);
}
```

```
return 0;
```

```
}
```

<https://onlinegdb.com/1yZe6yZsu>



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
EXAMEN DE: FUNDAMENTOS DE PROGRAMACIÓN
FORMA A PRACTICA

II
OP2e

PERÍODO:
NOMBRE: Edersun Gualuto
CARRERA: Ing. Electrónica y Automatización PARCIAL: 3
CURSO (NRC): 29543
FECHA: 21/01/2026

INDICACIONES GENERALES:

- La evaluación es personal y no se permite intercambiar información con sus compañeros, si es llamado la atención por una vez, se procederá a ponerle CERO y no recibirá su evaluación.
- La parte práctica debe subirlo en la carpeta de GitHub y formato indicado por el Sr. Docente (Apellidos Nombres_U2_Examen).
- Dispone de 2h.

Parte Práctica (20 Puntos)

Instrucciones.

1. A continuación, se presentan los requisitos funcionales (RF) para el desarrollo de su programa.
2. Recuerde que en el código elaborado por Ud. Debe comentar donde desarrolla el RF solicitado, en este documento coloque el código y al final las capturas de pantalla que evidencien los RF.
3. Desde ya éxitos en su evaluación.
4. Este documento debe ser subido en formato PDF con Apellidos Nombres_U2_Examen, a la tarea al AULA 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 terminar, el programa debe mostrar una tabla con cada intento y resultado textual, y revelar el número secreto si no se adivinó.

2. RUBRICA DE CALIFICACIÓN

Criterio	4 puntos – Excelente	3 puntos – Bueno	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.	D
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.	D

Uso de la matriz para almacenar datos	Nueva implementación correctamente y se imprime al final.	Nueva implementación, pero con errores menores.	Uso parcial de la matriz.	No se usa la matriz adecuadamente.	O
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.	1
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.	L
TOTAL, SOBRE 20 PUNTOS					

Elaborado por: Ing Jenny A Ruiz R
 Docente TC DCCO
 Fecha: 21/01/2026

Nombrei Ederson Gualoto

Fecha: 21/01/2026

NRC: 29583

Esquema
en pseudocódigo

Evaluación Parcial

Tabla de datos

Objeto	Nombre	Variable	Tipo
Rango	Min	número	Variable
Rango	Max	número	Variable
Validación	Sí	0 - 10	Caracter
Validación	Sí no	-1 o 11 - ∞	Condición
Intento	Registro	intento	Condición
Valor	Ingresado	valor	Caracter
resultado	resultado	valor	Caracter
Tabla	Tabla	Resumen datos	Caracter
Terminar	Fin	Terminar	Caracter

Esquema en pseudocódigo

// Definir Min y Max /

Escribir (Ingresar valor mínimo);

Leer Min;

Escribir (Ingresar valor máximo);

Leer Max; /

// numero de intentos

Escribir (Ingresar rango de 0 a 10);

Leer X; /

// Bucle de X

Escribir (Ingresar número);

Leer Y;

Si X >