

## CORRECCION DEL CODIGO “NUMERO SECRETO”

Nombre: Dawin Tapia

Fecha: 23/01/2026

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

int main() {
    int min, max, intentos; // Variables para el rango y cantidad de intentos
    int secreto;           // Número secreto que se debe adivinar
    int i, j;              // Contadores para los bucles
    int numero;            // Número ingresado por el usuario
    int repetido;          // Verificar números repetidos
    int adivinado = 0;      // Saber si el número fue adivinado

    printf("=====Bienvenido al juego del numero secreto=====\\n");

    // Sigue el rango mínimo
    printf("Ingrese el rango minimo: ");
    scanf("%d", &min);

    // Sigue el rango máximo
    printf("Ingrese el rango maximo: ");
    scanf("%d", &max);

    // Verifica que el rango sea válido
    if (min >= max) {
        printf("Rango invalido.\\n");
        return 0;           // Termina el programa si el rango es incorrecto
    }
```

```
// Solicitud de la cantidad de intentos
printf("Diga un numero para sus intentos: ");
scanf("%d", &intentos);

// Verifica que los intentos sean válidos
if (intentos <= 0) {
    printf("Cantidad de intentos invalida.\n");
    return 0;
}

// Genera el número secreto dentro del rango indicado
secreto = rand() % (max - min + 1) + min;

int numeros[intentos];      // Guardar los números ingresados
char resultado[intentos][10]; // Guardar el resultado de cada intento

// Verifica que los intentos sean menores a 10
if (intentos >= 10) {
    printf("Número de intentos menor a 10.\n");
    return 0;
}

// Bucle principal del juego
for (i = 0; i < intentos; i++) {

    // Muestra cuántos intentos quedan
    printf("\nIntentos restantes: %d\n", intentos - i);
    printf("Ingrese un numero: ");
    scanf("%d", &numero);

    // Validación del rango del número ingresado
```

```
if (numero < min || numero > max) {
    printf("Número fuera del rango.\n");
    i--;
    // No cuenta el intento
    continue;
    // Vuelve a pedir el número
}

// Validación de número repetido
repetido = 0;
for (j = 0; j < i; j++) {
    if (numeros[j] == numero) {
        repetido = 1; // Marca como repetido
        break;
    }
}

// Si el número ingresado es repetido
if (repetido) {
    printf("Número repetido. Intente otro.\n");
    i--;
    // No cuenta el intento
    continue;
}

// Guarda el número ingresado
numeros[i] = numero;

// Compara el número con el numero secreto
if (numero < secreto) {
    printf("El numero es muy bajo.\n");
    sprintf(resultado[i], "Muy bajo"); // Guarda el resultado
} else if (numero > secreto) {
    printf("El numero es muy alto.\n");
}
```

```

        sprintf(resultado[i], "Muy alto"); // Guarda el resultado
    } else {
        // Si el número es correcto
        printf("=====¡CORRECTO!Adivinaste el numero=====\\n");
        sprintf(resultado[i], "Correcto"); // Guarda el resultado
        adivinado = 1; // Marca que fue adivinado
        i++; // Incrementa para mostrar correctamente la tabla
        break; // Sale del bucle
    }
}

// Muestra la tabla de resultados
printf("\\n--- TABLA DE INTENTOS ---\\n");
printf("Intento\tNúmero\tResultado\\n");

// Recorre y muestra cada intento
for (j = 0; j < i; j++) {
    printf("%d\t%d\t%s\\n", j + 1, numeros[j], resultado[j]);
}

// Si no se adivinó el número secreto
if (!adivinado) {
    printf("\\nNo adivinaste el numero.\\n");
    printf("El numero secreto era: %d\\n", secreto);
}

return 0; // Fin del programa
}

```

GDBonline link:  
<https://onlinegdb.com/oFSK7Yv6E>