



Programas
Académicos



Actividad 3 - Código en lenguaje C

Introducción al desarrollo de software

Ingeniería en Desarrollo de Software.

Tutor: Felipe de Jesús Araux López

Alumno:

Luis José Monroy Rodríguez G2 Luisjo2000mr@hotmail.com

Fecha: 30/07/2023

Índice

Introducción	2
Descripción	3
Justificación.....	4
Desarrollo	5
Codificación	5
Primos	5
Par/Impar.....	7
Al Revés	9
Ejecución en el compilador	10
Primos	10
Par/Impar.....	11
Al Revés	11
Conclusión.....	12
Referencias	13

Introducción


Recordemos que, C es un lenguaje de programación de propósito general originalmente desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell, como evolución del anterior lenguaje B, a su vez basado en BCPL, al igual que B, es un lenguaje orientado a la implementación de sistemas operativos, concretamente Unix, C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas y aplicaciones, las principales características del lenguaje C llevan a cabo un conjunto completo de instrucciones de control, la agrupación de instrucciones, el concepto de puntero y la transferencia de argumentos de funciones por valor, además, se caracteriza por su minimalismo, ya que fue pensado como una herramienta sencilla para escribir programas que se compilan con un compilador, tomando en cuenta lo anterior, procederemos a continuación a observar la siguiente actividad que mostrará el lenguaje C llevado a la práctica.

Descripción

En la presente actividad, se llevará a cabo la realización de la codificación en lenguaje C de las 3 calculadoras, Primos, Par/Impar y Al revés, en base a la previa construcción de sus respectivos diagramas de flujo, será llevado a cabo tomando en cuenta cada detalle del código y explicándolo de manera clara y precisa para que el lector al revisar el documento, tenga la seguridad y satisfacción de que tendrá un aprendizaje integro y de una manera un poco más sencilla de lo habitual, ese es el propósito de la presente actividad, para que de esa manera se reconozca la importancia de todos los pasos previos a seguir para la correcta codificación de un programa y así ser más ágiles al tratarse de realizar cualquier solución a cualquier tipo de problema que se presente en el futuro, a continuación se mostrará evidencia de cada código en Lenguaje C ejecutado en Programiz para demostrar que funciona, posteriormente se explicará la lógica utilizada en cada código.

Justificación

Cabe destacar que, C es un lenguaje de programación de propósito general que ofrece una programación estructurada y un conjunto completo de instrucciones de control, lo que lo hace ideal para desarrollar programas como los descritos en la actividad, además, C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas y aplicaciones, esto significa que los programas desarrollados en C serán rápidos y eficientes, además, C es un lenguaje ampliamente utilizado y cuenta con una gran comunidad de desarrolladores y recursos en línea, lo que facilita el aprendizaje y la resolución de problemas durante el desarrollo de los programas, en resumen, el lenguaje de programación C es una excelente opción para desarrollar los programas de calculadora descritos en la actividad debido a su capacidad para ofrecer una programación estructurada, su eficiencia y su amplia adopción y disponibilidad de recursos.



Desarrollo

Codificación

Primos

```
#include <stdio.h>

int main() {

    int numero, cont = 0;

    printf("Ingresa un número: ");

    scanf("%d", &numero);

    for (int i = 1; i <= numero; i++) {

        if (numero % i == 0) {

            cont++;

        }

    }

    if (cont > 2) {

        printf("Este número no es primo\n");

    } else {

        printf("Este número es primo\n");

    }

    return 0;

}
```

Primero, se pide al usuario que ingrese un número entero y se almacena en la variable “numero”.

Luego, se inicializa una variable llamada “cont” en 0.

Después, se ejecuta un ciclo para cada número entero “i” desde 1 hasta “numero”.

Dentro del ciclo, se verifica si “numero” es divisible por “i”. Si es así, se incrementa la variable “cont” en 1.

Una vez que el ciclo ha terminado, se verifica si la variable “cont” es mayor a 2. Si es así, se imprime que el número no es primo.

Si no, se imprime que el número es primo.


Cabe destacar que la variable numero almacena el número entero ingresado por el usuario, se utiliza para verificar si el número es primo o no.

cont: Esta variable se inicializa en 0 y se utiliza para contar cuántos divisores tiene el número ingresado por el usuario. Si el número de divisores es mayor a 2, entonces el número no es primo.

i: Esta variable se utiliza como contador en el ciclo for. Se inicializa en 1 y se incrementa en 1 en cada iteración del ciclo hasta que alcanza el valor de numero.

printf: Esta función se utiliza para imprimir mensajes en la pantalla. En este caso, se utiliza para pedir al usuario que ingrese un número y para imprimir si el número es primo o no.

scanf: Esta función se utiliza para leer datos ingresados por el usuario. En este caso, se utiliza para leer el número entero ingresado por el usuario y almacenarlo en la variable numero.



Par/Impar

```
#include <stdio.h>

int main() {

    int numeros[10];

    for (int i = 0; i < 10; i++) {

        printf("Ingresa un número: ");

        scanf("%d", &numeros[i]);

    }

    for (int i = 0; i < 10; i++) {

        if (numeros[i] % 2 == 0) {

            printf("%d este número es par\n", numeros[i]);

        } else {


            printf("%d este número es impar\n", numeros[i]);

        }

    }

    return 0;


}
```



Primero, se inicializa un arreglo de enteros llamado “numeros” para almacenar 10 números.

Luego, se ejecuta un ciclo for para pedir al usuario que ingrese 10 números enteros. En cada iteración del ciclo, se utiliza la función scanf para leer el número ingresado por el usuario y se almacena en el arreglo “numeros”.

Después, se ejecuta otro ciclo for para recorrer cada número en el arreglo “numeros”. En cada iteración del ciclo, se verifica si el número es divisible por 2 utilizando el operador módulo (%). Si es así, se imprime que el número es par utilizando la función printf. Si no, se imprime que el número es impar.



Al Revés

```
#include <stdio.h>

int main() {

    int numero, invertido = 0;

    printf("Ingresa un número de 4 dígitos: ");

    scanf("%d", &numero);

    while (numero > 0) {

        invertido = invertido * 10;

        invertido = invertido + numero % 10;

        numero = numero / 10;

    }

    printf("El número al revés es: %d\n", invertido);

    return 0;

}
```

Primero, se pide al usuario que ingrese un número entero de 4 dígitos y se almacena en la variable “numero”.

Luego, se inicializa una variable llamada “invertido” en 0.

Después, se ejecuta un ciclo while mientras el valor de “numero” sea mayor que 0. En cada iteración del ciclo, se multiplica “invertido” por 10, se suma a “invertido” el último dígito de “numero” y se elimina el último dígito de “numero” dividiéndolo entre 10.

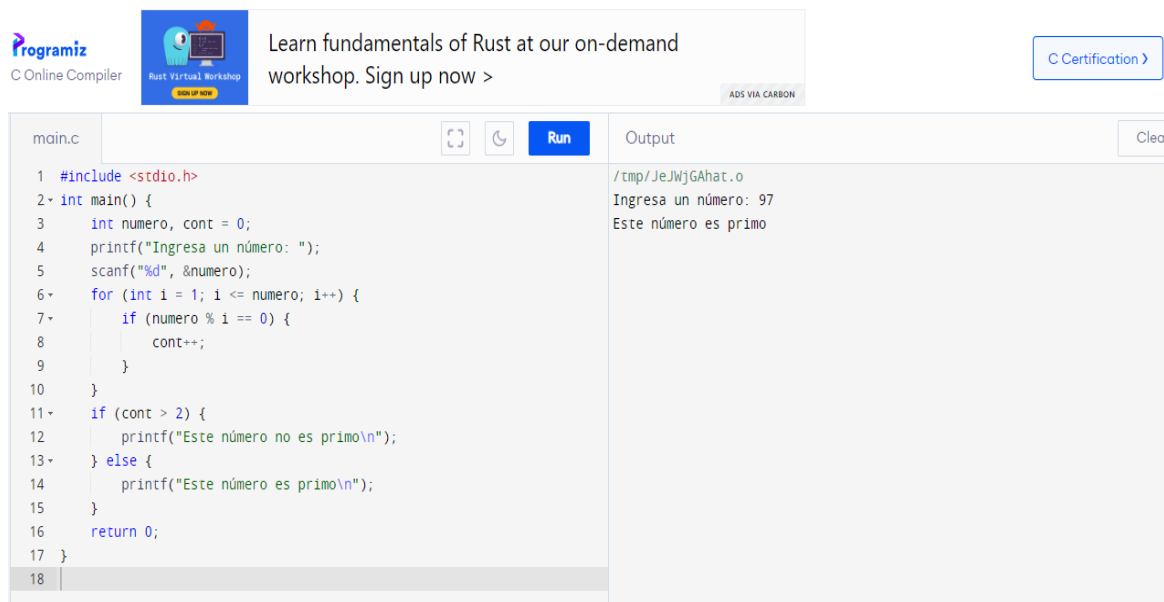
Una vez que el ciclo ha terminado, se imprime el valor de “invertido”.

Cabe destacar que un ciclo while es una estructura de control de flujo en programación que permite repetir un bloque de código mientras se cumpla una condición.

Ejecución en el compilador

Primos

Figura 1



The screenshot shows the Programiz C Online Compiler interface. At the top, there is a banner for 'Learn fundamentals of Rust at our on-demand workshop. Sign up now >' with a 'C Certification >' button. Below the banner, the compiler window is titled 'main.c'. The code is as follows:

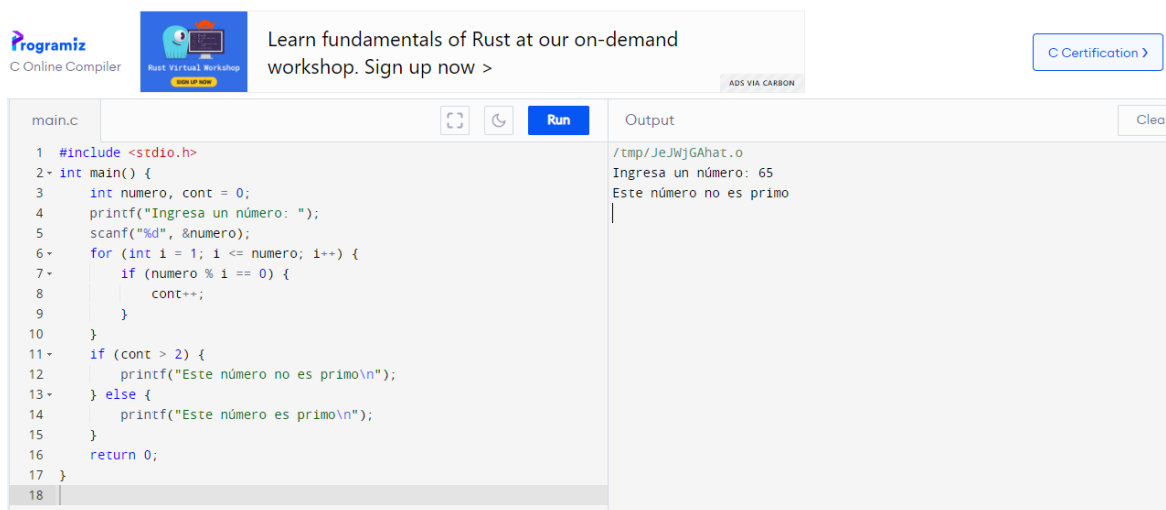
```
1 #include <stdio.h>
2 int main() {
3     int numero, cont = 0;
4     printf("Ingresa un número: ");
5     scanf("%d", &numero);
6     for (int i = 1; i <= numero; i++) {
7         if (numero % i == 0) {
8             cont++;
9         }
10    }
11    if (cont > 2) {
12        printf("Este número no es primo\n");
13    } else {
14        printf("Este número es primo\n");
15    }
16    return 0;
17 }
```

The 'Run' button is highlighted. The 'Output' window on the right shows the following text:

```
/tmp/JeJWjGAhat.o
Ingresa un número: 97
Este número es primo
```

Nota: Creación propia, muestra que el numero ingresado es primo.

Figura 2



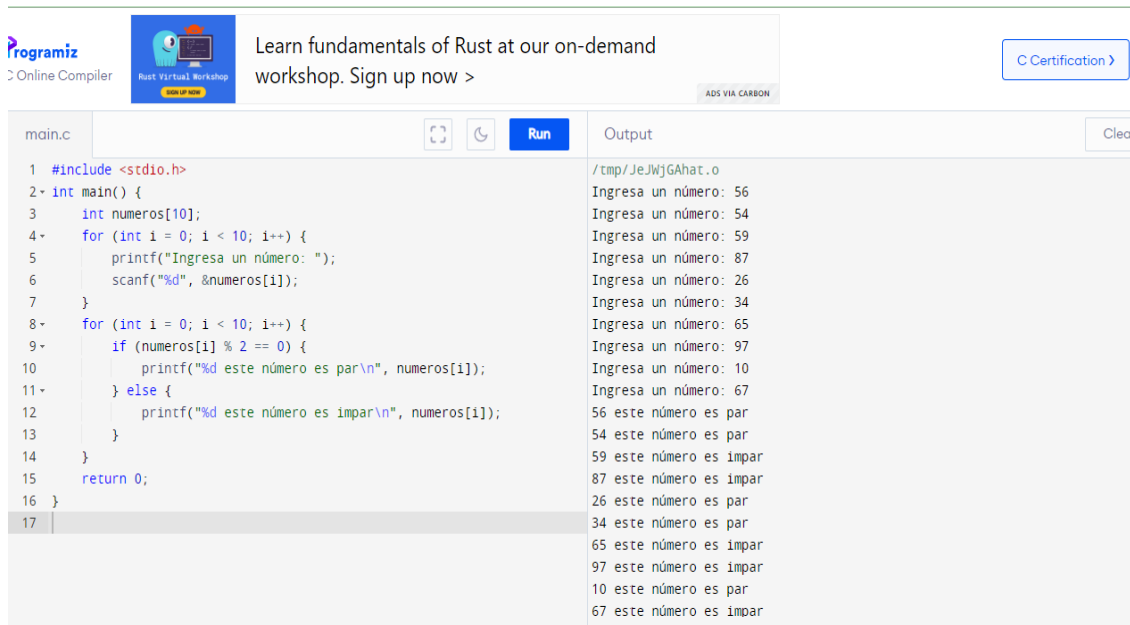
The screenshot shows the same Programiz C Online Compiler interface as Figure 1. The code is identical. The 'Run' button is highlighted. The 'Output' window on the right shows the following text:

```
/tmp/JeJWjGAhat.o
Ingresa un número: 65
Este número no es primo
```

Nota: Creación propia, muestra que el numero ingresado es no primo.

Par/Impar

Figura 3



Programiz
C Online Compiler

Learn fundamentals of Rust at our on-demand workshop. Sign up now >

ADS VIA CARBON

main.c

```

1 #include <stdio.h>
2 int main() {
3     int numeros[10];
4     for (int i = 0; i < 10; i++) {
5         printf("Ingresa un número: ");
6         scanf("%d", &numeros[i]);
7     }
8     for (int i = 0; i < 10; i++) {
9         if (numeros[i] % 2 == 0) {
10             printf("%d este número es par\n", numeros[i]);
11         } else {
12             printf("%d este número es impar\n", numeros[i]);
13         }
14     }
15     return 0;
16 }
17

```

Run

Output

Clear

```

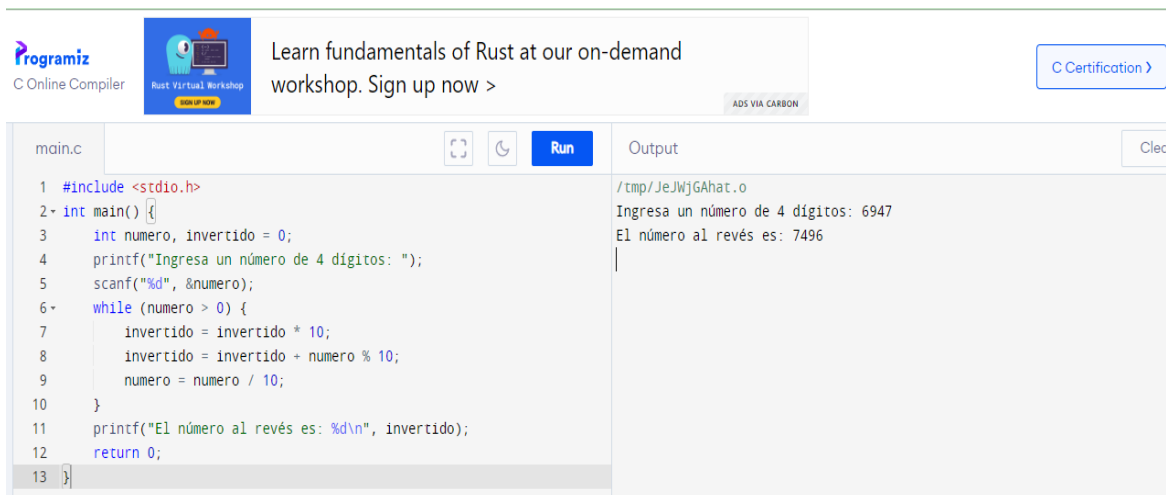
/tmp/JeJWjGAhat.o
Ingresa un número: 56
Ingresa un número: 54
Ingresa un número: 59
Ingresa un número: 87
Ingresa un número: 26
Ingresa un número: 34
Ingresa un número: 65
Ingresa un número: 97
Ingresa un número: 10
Ingresa un número: 67
56 este número es par
54 este número es par
59 este número es impar
87 este número es impar
26 este número es par
34 este número es par
65 este número es impar
97 este número es impar
10 este número es par
67 este número es impar

```

Nota: Creación propia, muestra que de los diez números ingresados cinco números son pares ingresados son pares y los otros cinco son impares.

Al Revés

Figura 4



Programiz
C Online Compiler

Learn fundamentals of Rust at our on-demand workshop. Sign up now >

ADS VIA CARBON

main.c

```

1 #include <stdio.h>
2 int main() {
3     int numero, invertido = 0;
4     printf("Ingresa un número de 4 dígitos: ");
5     scanf("%d", &numero);
6     while (numero > 0) {
7         invertido = invertido * 10;
8         invertido = invertido + numero % 10;
9         numero = numero / 10;
10    }
11    printf("El número al revés es: %d\n", invertido);
12    return 0;
13 }

```

Run

Output

Clear

```

/tmp/JeJWjGAhat.o
Ingresa un número de 4 dígitos: 6947
El número al revés es: 7496

```

Nota: Creación propia, muestra el número al revés de cuatro dígitos ingresado.

Conclusión

La actividad presentada, que consiste en el desarrollo de tres programas de calculadora utilizando el lenguaje de programación C, es importante tanto en el campo laboral como en la vida cotidiana. Estos programas permiten a los usuarios realizar cálculos y operaciones matemáticas de manera rápida y eficiente, lo que puede ser útil en una amplia variedad de contextos, en el campo laboral, estos programas pueden ser utilizados para realizar cálculos y análisis de datos en áreas como la ingeniería, la ciencia, la economía y las finanzas. Por ejemplo, el programa Primos puede ser utilizado para identificar números primos en conjuntos de datos, lo que puede ser útil en criptografía y seguridad informática. El programa Par/Impar puede ser utilizado para analizar conjuntos de datos y determinar si hay una mayor cantidad de números pares o impares, lo que puede ser útil en estadística y análisis de datos. El programa Al Revés puede ser utilizado para invertir números y realizar cálculos con ellos, lo que puede ser útil en matemáticas y ciencias, en la vida cotidiana, estos programas también pueden ser útiles para realizar cálculos y operaciones matemáticas rápidamente. Por ejemplo, el programa Primos puede ser utilizado para verificar si un número es primo o no, lo que puede ser útil para resolver problemas matemáticos o jugar juegos de lógica. El programa Par/Impar puede ser utilizado para determinar si un conjunto de números es mayormente par o impar, lo que puede ser útil para tomar decisiones o resolver problemas. El programa Al Revés puede ser utilizado para invertir números y realizar cálculos con ellos, lo que puede ser útil para resolver problemas matemáticos o jugar juegos de lógica.

En resumen, la actividad presentada es importante tanto en el campo laboral como en la vida cotidiana ya que permite a los usuarios realizar cálculos y operaciones matemáticas de manera rápida y eficiente utilizando programas desarrollados en el lenguaje de programación C.

Referencias

—ICTEA. (s.f.). ¿Qué es el lenguaje de programación C? Base de Conocimientos - ICTEA.

Recuperado de <https://www.ictea.com/cs/index.php?rp=%2Fknowledgebase%2F8834%2FiQue-es-el-lenguaje-de-programacion-C.html>

—Programacion218. (2016, 16 de abril). Características principales del lenguaje C.

WordPress. Recuperado de <https://programacion218.wordpress.com/2016/04/16/caracteristicas-principales-de-lenguaje-c/>