

Actividad | # 2 | Código en lenguaje C

Introducción al Desarrollo de Software

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Dévora

ALUMNO: Genaro Kantun Can

FECHA: 20-04-2025

Índice

Introducción.....	3
Descripción.....	4
Justificación.....	5
Desarrollo.....	6
Calculadora Par/Impar.....	6
Explicación calculadora Par/Impar.....	7
Calculadora número primo.....	9
Explicación calculadora número primo.....	10
Calculadora al revés.....	12
Explicación calculadora al revés.....	13
Conclusión.....	15
Referencias.....	16
Link GitHub.....	16

Introducción

El lenguaje de programación C, es uno de los lenguajes mas utilizados en la historia de la informática, desarrollado por Dennis Ritchie entre 1969 y 1972, como evolución del anterior lenguaje B, se caracteriza por su eficiencia, flexibilidad y portabilidad, es apreciado por la eficiencia del código que produce y es el lenguaje de programación mas popular para crear software de sistemas y aplicaciones.

Al igual que en las actividades pasadas, seguimos con los ejercicios de las calculadoras, pero en esta ocasión, se va a programar utilizando con compilador de lenguaje C, cada una de estas calculadoras tienen una función en específico, la primera se trata de programar para que nos muestre si un número ingresado es primo o no.

La segunda calculadora nos mostrará si los números que ingresemos son pares o impares, y la tercera calculadora ingresaremos un número y esta nos permitirá visualizar el número ingresado, pero al revés, es decir, si ingresamos el número 1235, esta nos devolverá el siguiente resultado 5321.

"Medir el progreso del desarrollo de software por líneas de código es como medir el progreso de la construcción de un avión por su peso" (Bill Gates, S.F.)

Descripción

Para la siguiente actividad, seguimos con los ejercicios de las calculadoras de las actividades pasadas, las cuales tiene como objetivo ayudar a colegios y escuelas públicas, cada una cuenta con una función en específico, pero en esta ocasión se programarán utilizando un compilador en lenguaje C.

La primera calculadora tendrá como nombre primo, y su función será que al ingresar un número esta nos mande un mensaje donde nos diga si el número ingresado es primo o no.

La segunda tendrá por nombre par/impar, y su función será que al ingresar un número nos diga si es un número par o es un número impar.

Y la tercera tendrá por nombre al revés, esta tendrá como función que, al ingresar un número, esta nos devuelva el número, pero al revés, por ejemplo, si nosotros ingresamos el número 1235, esta nos muestre lo siguiente 5321.

Al realizar la programación aprenderemos sobre el uso de un compilador en lenguaje C y la importancia que este tiene en el mundo de la informática y la programación.

Justificación

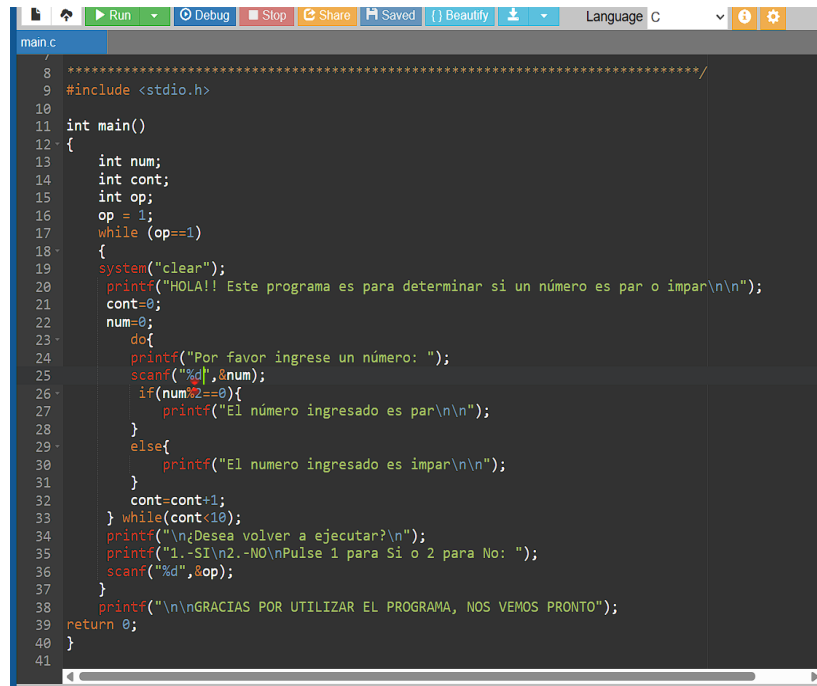
Aprender a programar puede abrir puertas a oportunidades de trabajo y capacitar a personas para participar en la creación y el desarrollo de tecnología, la programación es un componente fundamental en la era digital moderna, este lenguaje es la herramienta principal para fomentar habilidades, resolución de problemas y brindar herramientas para enfrentar desafíos.

Al realizar la presente actividad, tendremos la noción de cómo utilizar los diferentes códigos, los ciclos y las funciones que tiene cada uno de ellos; con los ejercicios presentados en esta actividad, se pretende utilizar diferentes códigos o condiciones, para que cada calculadora realice la función solicitada.

Aprenderemos la importancia del lenguaje de programación C, ya que no ayudará en un futuro, ya que es la base para otros lenguajes como son C++ y Java, por lo que entender C puede facilitar el aprendizaje de estos lenguajes, hoy en día la programación no solo es para personas que trabajan en la tecnología, sino que esta presente en todo lo que utilizamos.

Desarrollo

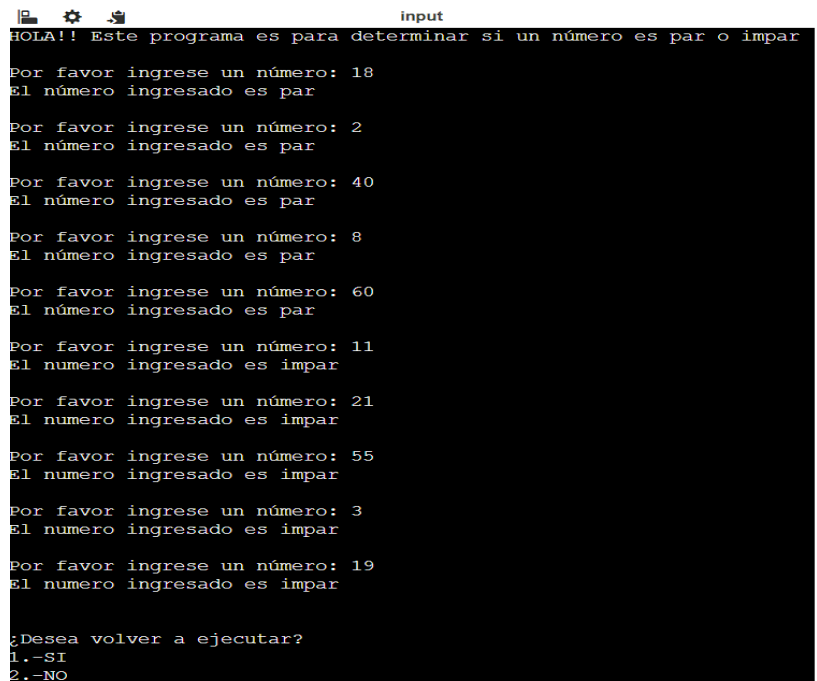
Calculadora Par/Impar



```

main.c
8  /*
9  #include <stdio.h>
10
11 int main()
12 {
13     int num;
14     int cont;
15     int op;
16     op = 1;
17     while (op==1)
18     {
19         system("clear");
20         printf("HOLA!! Este programa es para determinar si un número es par o impar\n\n");
21         cont=0;
22         num=0;
23         do{
24             printf("Por favor ingrese un número: ");
25             scanf("%d",&num);
26             if(num%2==0){
27                 printf("El número ingresado es par\n\n");
28             }
29             else{
30                 printf("El numero ingresado es impar\n\n");
31             }
32             cont=cont+1;
33         } while(cont<10);
34         printf("\n¿Desea volver a ejecutar?\n");
35         printf("1.-SI\n2.-NO\nPulse 1 para Si o 2 para No: ");
36         scanf("%d",&op);
37     }
38     printf("\n\nGRACIAS POR UTILIZAR EL PROGRAMA, NOS VEMOS PRONTO");
39     return 0;
40 }
41
  
```

Fig. #1



```

input
HOLA!! Este programa es para determinar si un número es par o impar

Por favor ingrese un número: 18
El número ingresado es par

Por favor ingrese un número: 2
El número ingresado es par

Por favor ingrese un número: 40
El número ingresado es par

Por favor ingrese un número: 8
El número ingresado es par

Por favor ingrese un número: 60
El número ingresado es par

Por favor ingrese un número: 11
El numero ingresado es impar

Por favor ingrese un número: 21
El numero ingresado es impar

Por favor ingrese un número: 55
El numero ingresado es impar

Por favor ingrese un número: 3
El numero ingresado es impar

Por favor ingrese un número: 19
El numero ingresado es impar

¿Desea volver a ejecutar?
1.-SI
2.-NO
  
```

Fig. #2

Explicación calculadora Par/Impar

Esta calculadora es para determinar si un número es par o impar, primeramente, abrimos nuestro compilador, iniciamos nuestra biblioteca en este caso `#include <stdio.h>`, iniciamos nuestra función `“int main”`, a continuación, declaramos nuestras variables para esta ocasión van a ser de tipo entero `“int”` que representan lo siguiente:

`“Num”`, es el número que vamos a ingresar a nuestra calculadora.

`“Cont”`, es el conteo para de los números que el usuario ingrese, ya que la calculadora tendrá que leer como máximo 10 números.

`“Op”`, representa las opciones y la iniciamos con valor `=1` que es de asignación.

Iniciamos la instrucción `“while”` con nuestra variable `“op=1”`, en este caso es de condición, que es para que el programa se vuelva a ejecutar.

Para que la pantalla se borre escribimos la función `“system”` y entre paréntesis escribimos `“clear”` que es para que limpie o borre la pantalla, seguidamente escribimos la función `“printf”` para que nos imprima en pantalla un breve mensaje de lo que hace nuestra calculadora.

Iniciamos nuestra variable `“cont”` desde 0, también nuestra variable `“num”` de igual forma desde 0, a continuación, inicializamos nuestra instrucción `“do-while”`, para decirle al programa que vamos a repetir un bucle o un proceso.

Le decimos al usuario que `“ingrese un número”`, (como ya sabemos utilizando la función `“printf”`), seguidamente lo guardamos nuestra variable con la función `“scanf”`, le decimos que es de tipo entero con el formato `“%d”`, y lo guardamos con el formato `“&(ampersand)num”`.

Iniciamos nuestra condicional `“if”`, e iniciamos nuestra operación, el número que ingrese el usuario si su residuo al ser dividido entre 2 sea igual a cero, el programa va a imprimir el mensaje: `“el número ingresado es par”`, caso contrario, `“else”`, el siguiente mensaje será `“el número`

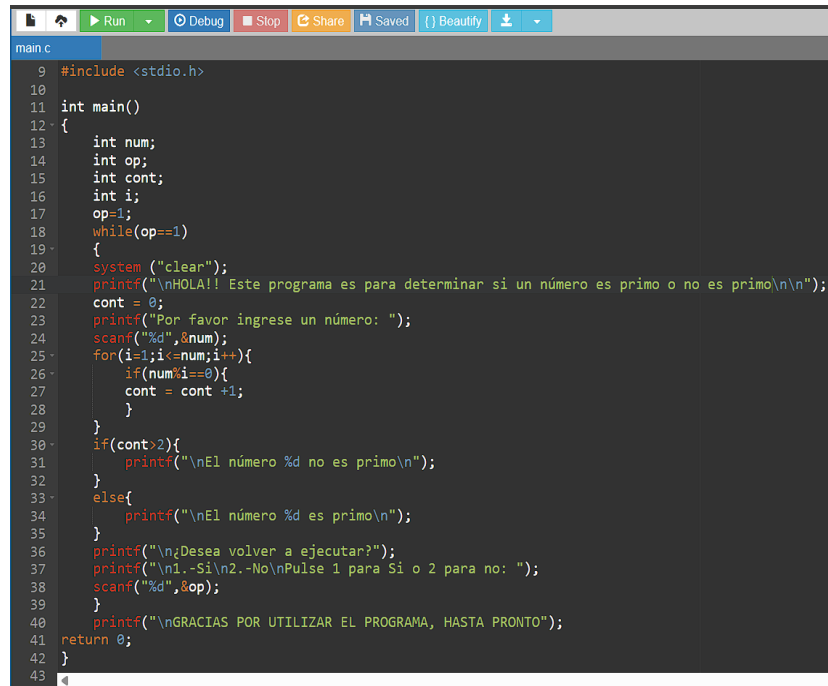
ingresado es impar”.

Iniciamos el contador que va a ir avanzando de 1 en 1, hasta llegar a 10 “while(<10)”, escribimos un mensaje haciendo una pregunta al usuario, si ¿ desea volver a ejecutar?, escribimos las opciones un 1 para si o un 2 para no y lo guardamos como entero con scanf “%d” en la variable “&op”. Si el usuario selecciona la opción 2 el programa le manda el siguiente mensaje de despedida: “gracias por utilizar el programa, nos vemos pronto” y con eso damos por terminado la programación de la calculadora “par/impar”

Adjunto link:

<https://www.onlinegdb.com/edit/1ugQac-hQ>

Calculadora números primos

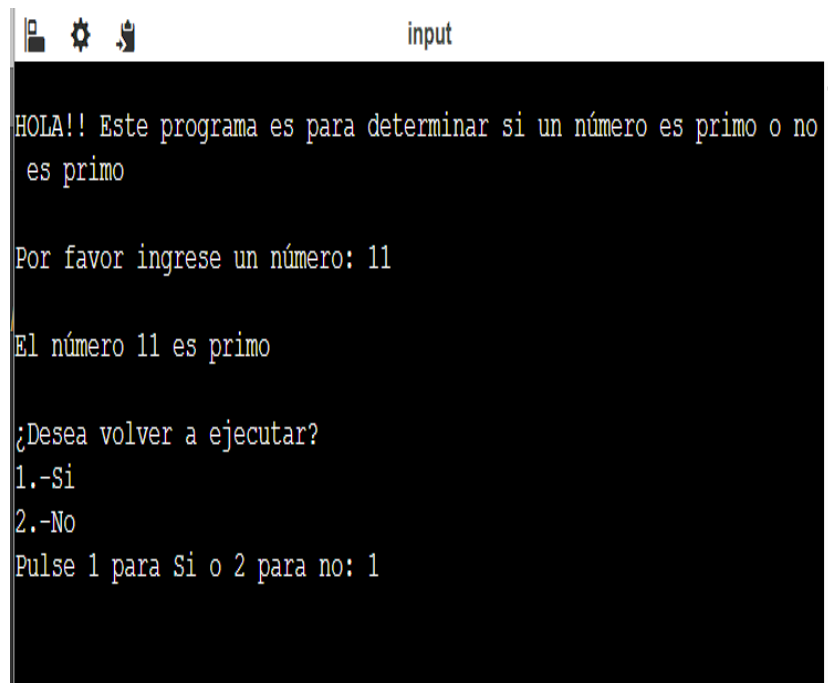


```

9  #include <stdio.h>
10
11 int main()
12 {
13     int num;
14     int op;
15     int cont;
16     int i;
17     op=1;
18     while(op==1)
19     {
20         system("clear");
21         printf("\nHOLA!! Este programa es para determinar si un número es primo o no es primo\n\n");
22         cont = 0;
23         printf("Por favor ingrese un número: ");
24         scanf("%d",&num);
25         for(i=1;i<=num;i++){
26             if(num%i==0){
27                 cont = cont +1;
28             }
29         }
30         if(cont>2){
31             printf("\nEl número %d no es primo\n");
32         }
33         else{
34             printf("\nEl número %d es primo\n");
35         }
36         printf("\n¿Desea volver a ejecutar?");
37         printf("\n1.-Si\n2.-No\nPulse 1 para Si o 2 para no: ");
38         scanf("%d",&op);
39     }
40     printf("\nGRACIAS POR UTILIZAR EL PROGRAMA, HASTA PRONTO");
41     return 0;
42 }
43

```

Fig. #3



```

input
HOLA!! Este programa es para determinar si un número es primo o no
es primo

Por favor ingrese un número: 11

El número 11 es primo

¿Desea volver a ejecutar?
1.-Si
2.-No
Pulse 1 para Si o 2 para no: 1

```

Fig. #4

```

input
HOLA!! Este programa es para determinar si un número es primo o no
es primo

Por favor ingrese un número: 6

El número 6 no es primo

¿Desea volver a ejecutar?
1.-Si
2.-No
Pulse 1 para Si o 2 para no: 2

```

Fig.#5

Explicación calculadora número primo

Esta calculadora es para determinar si un número es primo o no es primo.

Iniciamos nuestro compilador, iniciamos nuestra biblioteca en este caso `#include <stdio.h>`, iniciamos nuestra función “`int main`”, a continuación, declaramos nuestras variables para esta ocasión van a ser de tipo entero “`int`” que representan lo siguiente:

“`num`”, que va a representar el número que el usuario vaya a ingresar.

“`op`”, representa las opciones.

“`cont`”, que va a representar el contador.

“`i`”, nuestro iterador.

Asignamos valor a “`op`” que va a ser de asignación igual a 1, Iniciamos la instrucción “`while`” con nuestra variable “`op=1`”, en este caso es de condición, que es para que el programa se vuelva a ejecutar.

Para que la pantalla se borre escribimos la función “`system`” y entre paréntesis escribimos

“clear” que es para que limpie o borre la pantalla, seguidamente escribimos la función “printf” para que nos imprima en pantalla un breve mensaje de lo que hace nuestra calculadora.

Iniciamos el contador igual a cero, con la función “printf” le pedimos al usuario que “Por favor ingrese un número”, seguidamente lo guardamos con la función “scanf” %d &num, seguidamente iniciamos nuestra instrucción “for” donde nuestro iterador “i” va a estar iniciado en 1, también le asignamos que mientras sea menor o igual al número(num), que el usuario ingrese, y también le decimos que vaya aumentando de 1 en uno a nuestro contador.

Iniciamos nuestra condicional “if” donde hacemos nuestra operación, que es el número que ingrese el usuario(“num”), va a ser dividido por todos los números del bucle y su residuo sea cero, aumentamos el conteo de uno en uno.

Fuera del bucle iniciamos otra condicional “if” y hacemos la operación, si el conteo(“cont”) es mayor a 2, le decimos que nos imprima “printf” el siguiente mensaje “el número no es primo”, ya que el número que el usuario ingrese se divide con más de 2 números.

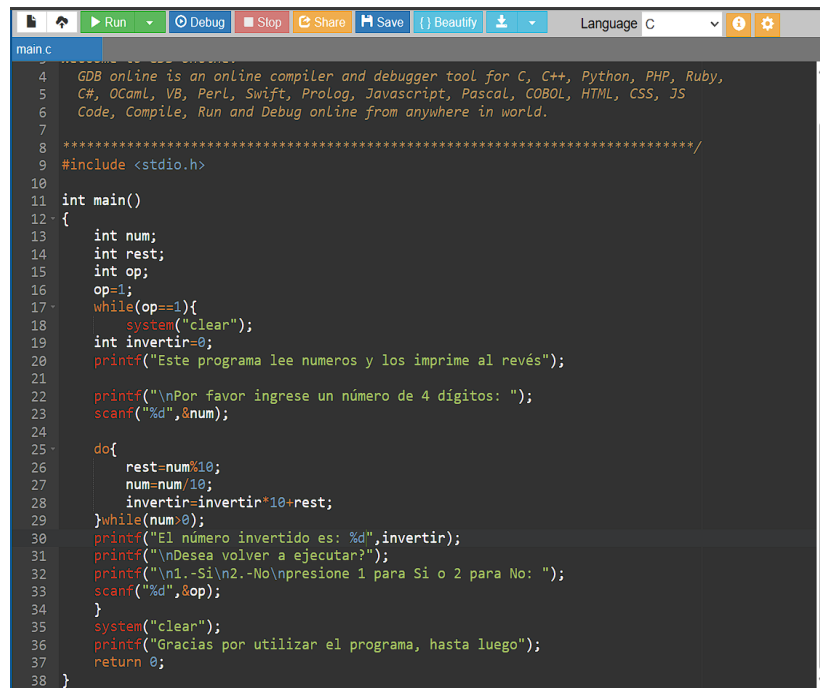
Caso contrario “else” el programa mandará el mensaje “el número es primo”, de igual forma le preguntamos al usuario si ¿Desea volver a ejecutar?, nuestras opciones 1 para Si o 2 para No, almacenamos los datos en “scanf %d, &op”. Si el usuario ya no desea volver a ejecutar el programa manda el siguiente mensaje: “gracias por utilizar el programa, hasta pronto”.

Con esto queda programada nuestra calculadora para determinar si un número es primo o no.

Adjunto link:

<https://www.onlinegdb.com/edit/qMWq648qk>

Calculadora al revés

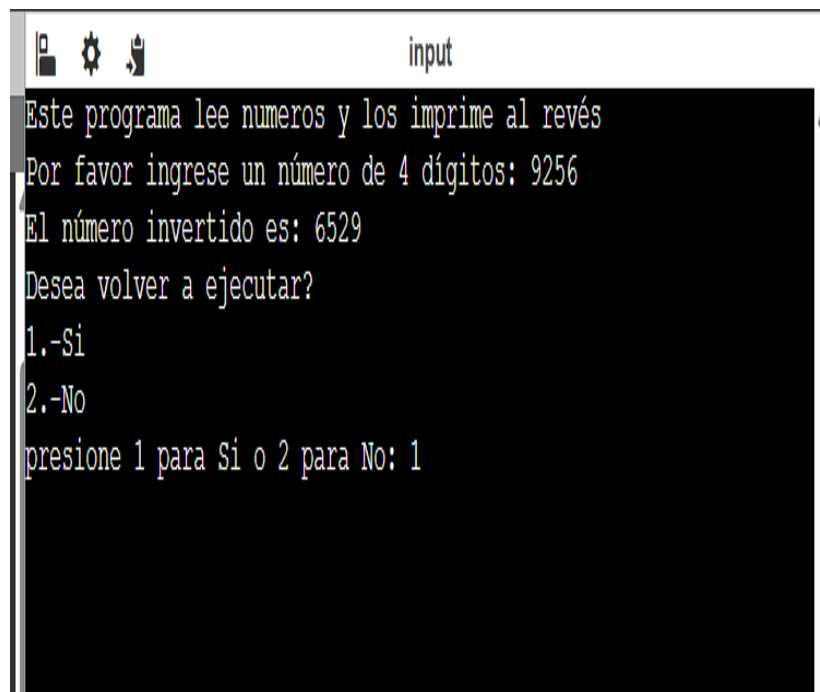


```

1  GDB online is an online compiler and debugger tool for C, C++, Python, PHP, Ruby,
2  C#, OCaml, VB, Perl, Swift, ProLog, Javascript, Pascal, COBOL, HTML, CSS, JS
3  Code, Compile, Run and Debug online from anywhere in world.
4
5  *****
6  #include <stdio.h>
7
8  int main()
9  {
10     int num;
11     int rest;
12     int op;
13     op=1;
14     while(op==1){
15         system("clear");
16         int invertir=0;
17         printf("Este programa lee numeros y los imprime al revés");
18
19         printf("\nPor favor ingrese un número de 4 dígitos: ");
20         scanf("%d",&num);
21
22         do{
23             rest=num%10;
24             num=num/10;
25             invertir=invertir*10+rest;
26         }while(num>0);
27         printf("El número invertido es: %d",invertir);
28         printf("\nDesea volver a ejecutar?");
29         printf("\n1.-Si\n2.-No\npresione 1 para Si o 2 para No: ");
30         scanf("%d",&op);
31     }
32     system("clear");
33     printf("Gracias por utilizar el programa, hasta luego");
34     return 0;
35 }

```

Fig. #6

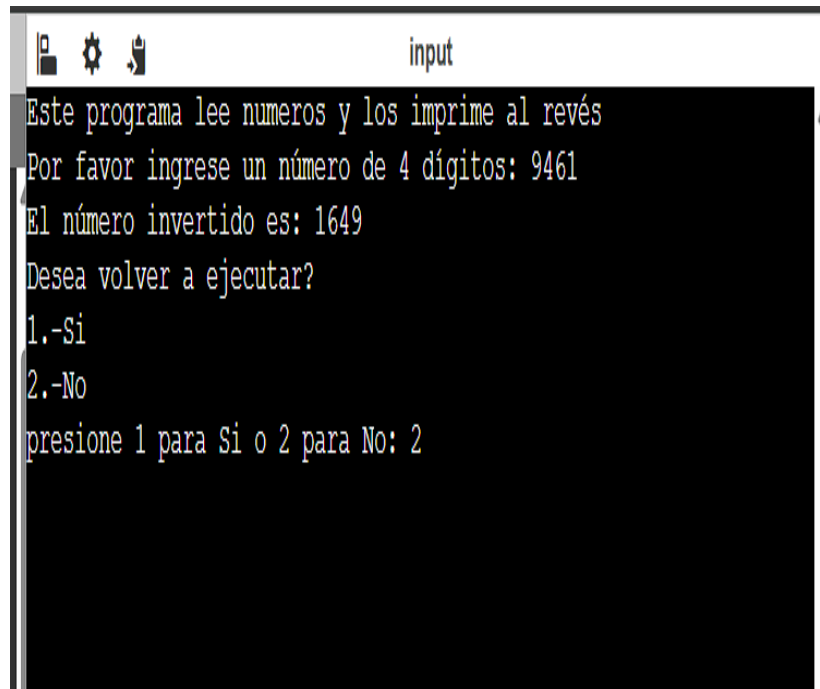


```

input
Este programa lee numeros y los imprime al revés
Por favor ingrese un número de 4 dígitos: 9256
El número invertido es: 6529
Desea volver a ejecutar?
1.-Si
2.-No
presione 1 para Si o 2 para No: 1

```

Fig. #7



```
Este programa lee numeros y los imprime al revés
Por favor ingrese un número de 4 dígitos: 9461
El número invertido es: 1649
Desea volver a ejecutar?
1.-Si
2.-No
presione 1 para Si o 2 para No: 2
```

Fig. #8

Explicación calculadora al revés

Este ejercicio trata de programar una calculadora que lea números de 4 dígitos y los imprima al revés.

Iniciamos nuestro compilador, iniciamos nuestra biblioteca en este caso `#include <stdio.h>`, iniciamos nuestra función “`int main`”, a continuación, declaramos nuestras variables para esta ocasión van a ser de tipo entero “`int`” que representan lo siguiente:

“`num`”, va a representar el número de 4 dígitos que el usuario ingrese.

“`rest`” representa el resto.

“`op`” representa las opciones.

Asignamos valor a “`op`” que va a ser de asignación igual a 1, Iniciamos la instrucción “`while`” con nuestra variable “`op=1`”, en este caso es de condición, que es para que el programa se vuelva a ejecutar.

Para que la pantalla se borre escribimos la función “system” y entre paréntesis escribimos “clear” que es para que limpie o borre la pantalla. Asignamos la variable invertir y la iniciamos en 0, esta variable va a representar el número ya al revés, seguidamente escribimos la función “printf” para que nos imprima en pantalla un breve mensaje de lo que hace nuestra calculadora.

Le pedimos al usuario con la función printf “por favor ingrese un número de 4 dígitos”, lo guardamos con la función scanf “%d &num”, e iniciamos la instrucción “do”, para posteriormente realizar nuestras operaciones.

Lo primero que debemos de hacer, es calcular el resto, por lo que operación queda así: “rest=num%10, para poder ir reduciendo nuestro número ingresado, la operación queda así: “num=num/10”, ahora para calcular nuestro número invertido la operación va a multiplicar el número invertido por 10 y va a sumar nuestro resto (“rest”), quedando la operación de la siguiente manera: “invertir=invertir*10+res”, cerramos nuestro ciclo “while”.

Ahora para mostrar en pantalla el resultado, iniciamos nuestra función “printf”, con el mensaje “el número invertido es (decimos que es un valor entero) “%d” y con nuestra variable “invertir”, le decimos al programa que nos muestre el número invertido.

Posteriormente el programa pregunta “¿desea volver a ejecutar?”. nuestras opciones 1 para Si o 2 para No, almacenamos los datos en “scanf %d, &op”. Si el usuario ya no desea volver a ejecutar el programa, le pedimos al programa que limpie la pantalla “system(clear) y seguidamente mandará el siguiente mensaje: “gracias por utilizar el programa, hasta luego”.

Con esto damos por terminado la programación de la calculadora para imprimir un número de 4 dígitos al revés.

Adjunto link:

https://www.onlinegdb.com/edit/muRci_kS7

Conclusión

Al realizar la presente actividad podemos concluir, que el lenguaje de programación C, debido a su eficiencia y flexibilidad es una herramienta valiosa, y es uno de los lenguajes de programación fundamental para la informática, al utilizarlo en el trabajo y en nuestra vida cotidiana podemos desarrollar soluciones eficientes y rápidas, crear programas y aplicaciones, y en lo personal puede ampliar nuestro conocimiento en programación.

Al utilizar el lenguaje de programación C, pude notar que es un lenguaje sencillo de aprender y requiere una continua práctica, y es una de las bases para otros lenguajes de programación como por ejemplo C++ y Java, así que una vez entendamos el lenguaje C, se nos facilitará entender los demás lenguajes.

En resumen, el lenguaje de programación C es una herramienta que nos ayuda a mejorar nuestra eficiencia y productividad en distintas áreas y el aprendizaje que proporciona puede ser un paso importante queremos mejorar nuestras habilidades en programación.

"Los mejores programadores no son sólo marginalmente mejores que los buenos. Se trata de un orden de magnitud mayor, medida por cualquier estándar: creatividad conceptual, velocidad, ingenio o habilidad para solucionar problemas" (Randall E. Stross. S.F)

Referencias

Wikipedia (2025,23 mar) C (Lenguaje de programación) Wikipedia la enciclopedia libre [https://es.wikipedia.org/wiki/C_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n)).

FESC (Cúcuta)(SF.) La importancia de la programación en la actualidad.
<https://www.fesc.edu.co/Revistas/OJS/index.php/convicciones/article/view/1394>

Joaquin10 (2010,10 de junio). Programación lenguaje C | Estructuras repetitivas | Programa para preguntar si se quiere repetir un ciclo. [video] YouTube.
<https://www.youtube.com/watch?v=adcKJUIJwoU>.

Link GitHub

<https://github.com/genkc/Introducci-n-al-desarrollo-de-software->