

Actividad | #3| Código en Lenguaje C

Introducción al Desarrollo de Software

Ingeniería en Desarrollo de Software

TUTOR: Sandra Luz Lara Devora

ALUMNO: Alejandra Ibarra Carmona

FECHA: 05/04/2024

Índice

Introducción ………………………………………………………

Descripción ………………………………………………………

Justificación ………………………………………………………

Desarrollo …………………………………………………………

Conclusión………………………………………………………

Referencias………………………………………………………

Introducción

En esta actividad se estarán presentando 3 calculadoras que a lo largo de 2 actividades anteriores pudimos programar, paso a paso cada una de ellas.

La librería stdio.h permite ingresar valores de entrada y salida, esta librería es la cabecera de un programa, ya que contiene constante declaraciones y macros para realizar operaciones básicas de entrada y salida. Trabaja ciertos caracteres como lo son datos y números y se ve reflejada en una tipa cadena para ejecutar se escribe de la siguiente manera como se ve en las capturas <stdio.h> y comenzamos a escribir el código, por otro lado, el macro se define de la siguiente manera #define el cual es capaz de realizar soluciones gráficas y matemáticas.

Otro punto importante dentro de nuestra librería son las funciones que se tienen que ejecutar y son algunas las que también aparecen en las capturas:

* getchar
* putchar
* scanf
* printf
* gets
* puts

Descripción

Es muy importante conocer que es un paradigma de programación y esto hace referencia a un conjunto de métodos de programación de un software que se encarga de clasificar a todos los lenguajes de programación y sus características. Cada uno de los lenguajes de programación cuenta con su paradigma según los tipos de ordenes que acepten, mencionaremos dos Paradigma Estructurado y Paradigma Orientado a Objetos.

Paradigma Estructurado se centra en mejorar la calidad y el tiempo de desarrollo de un programa y se caracteriza por no usar la instrucción goto, se pretende que con este paradigma resolver un problema de inicio a fin en una sola estructura o bloque de código.

Paradigma Orientado a Objetos es el mas utilizado en la actualidad, y permiten a los objetos operen a los datos que ingresas, con el fin de lograr obtener una salida de datos especifica, además muchas veces ya se encuentran prediseñados e incluidos en la biblioteca del lenguaje a utilizar.

Justificación

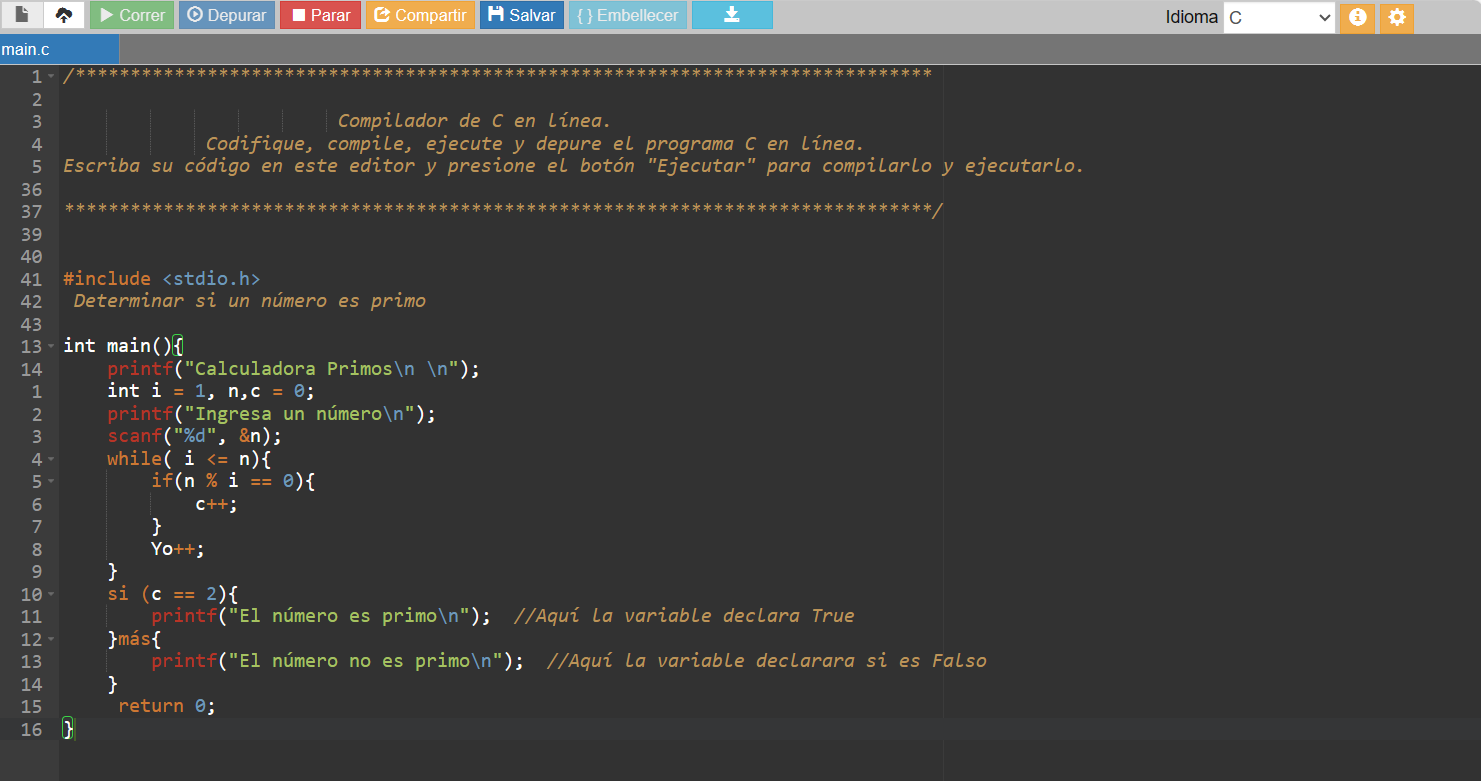
Saber crear diferentes tipos de algoritmos en cualquier lenguaje en fundamental porque se pueden implementar para algún requisito requerido en ese sistema, por decir si ese programa no acepta contraseñas con números pares, te dará más habilidad al momento de querer implementar nuevos códigos en otros lenguajes, en esta actividad de uso mucho las matemáticas, formulas y macros, hice prueba y error en varios códigos para ver si funcionaba como quería, se usaron algunos tipos de variables de bucle, hice las 3 calculadoras lo más completa y sencilla que pude para así sea entendible el código para su uso.

La programación se ah convertido en una habilidad esencial en la vida, desde el diseño de software y aplicaciones móviles hasta la automatización de procesos y análisis de datos.

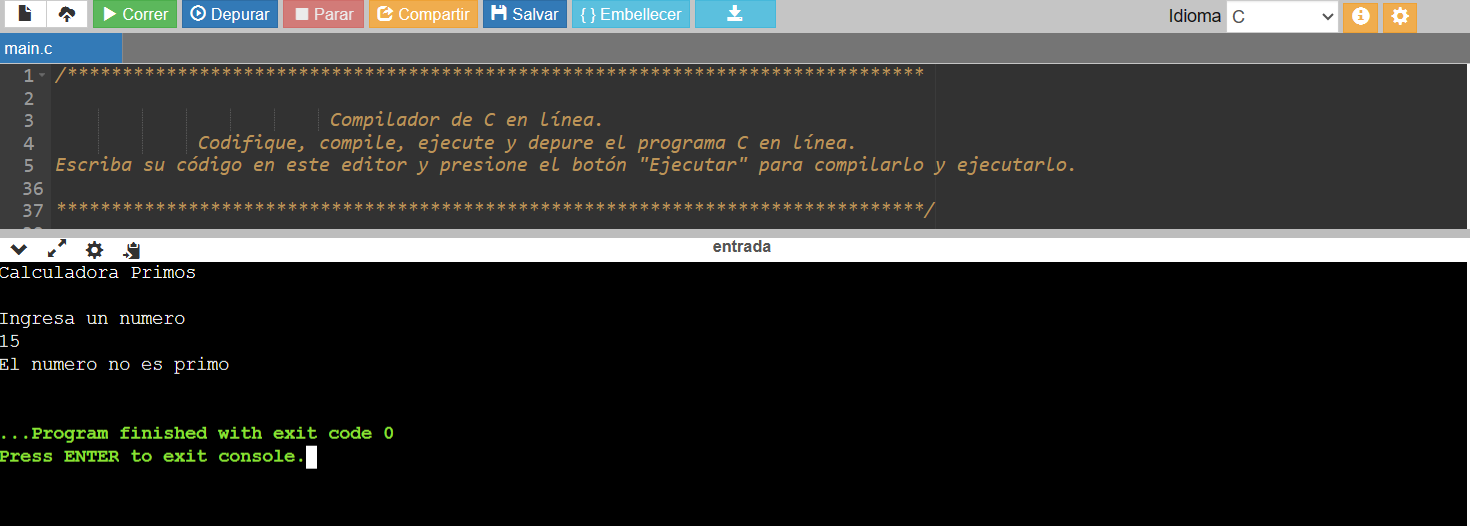
Desarrollo

Calculadora Primos: En esta calculadora su principal objetivo para el usuario es poder identificar si el numero que ingresa es primo o no es primo, esto con el fin de simplificar el trabajo del usuario así cuando ingrese cualquier tipo de numero en automático dará la respuesta como en el siguiente ejemplo se realizo primero con el numero 15 e imprimió que no es primo, mientras que con el numero 53 me indico que si es primo.

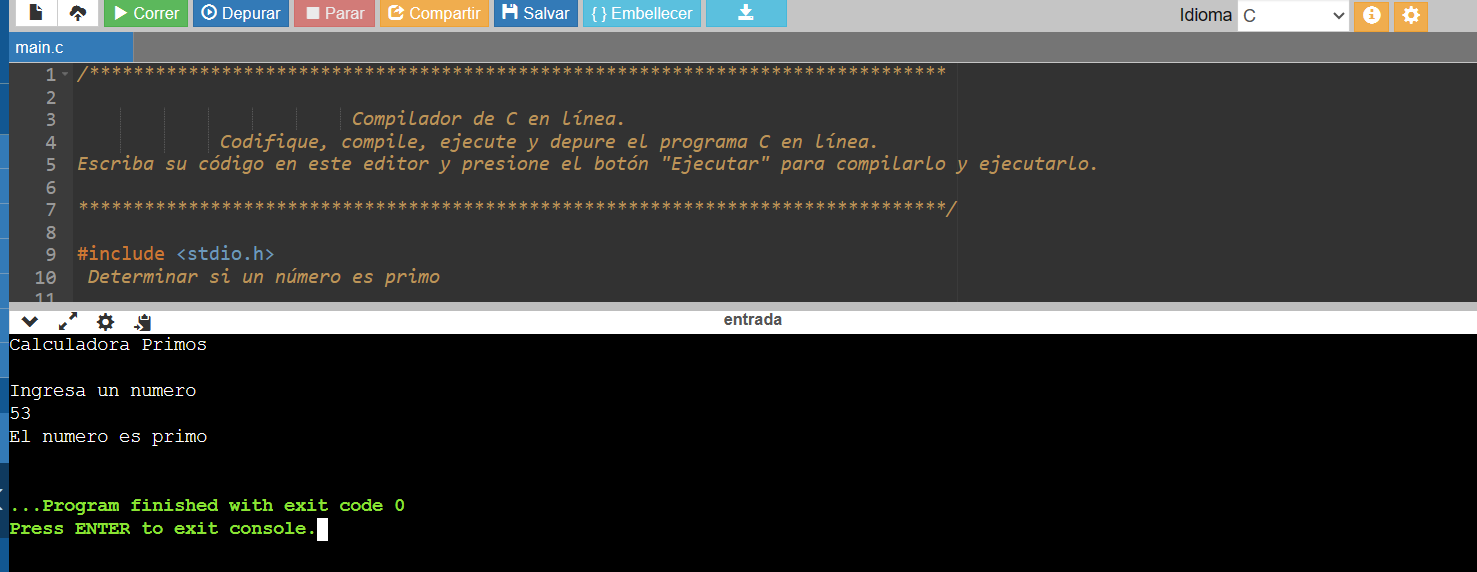
Imagen del código escrito en DEV++:



Shell del codigo, ejecutado dando como resultado “El numero no es primo”



Shell del codigo, ejecutado dando como resultado “El numero es primo”



Código de la calculadora Primos

#include <stdio.h>

//Determinar si un numero es primo

int main(){

printf("Calculadora Primos\n \n");

int i = 1, n,c = 0;

printf("Ingresa un numero\n");

scanf("%d", &n);

while( i <= n){

if(n % i == 0){

c++;

}

i++;

}

if (c == 2){

printf("El numero es primo\n"); //Aqui la variable declara True

}else{

printf("El numero no es primo\n"); //Aqui la variable declarara si es False

}

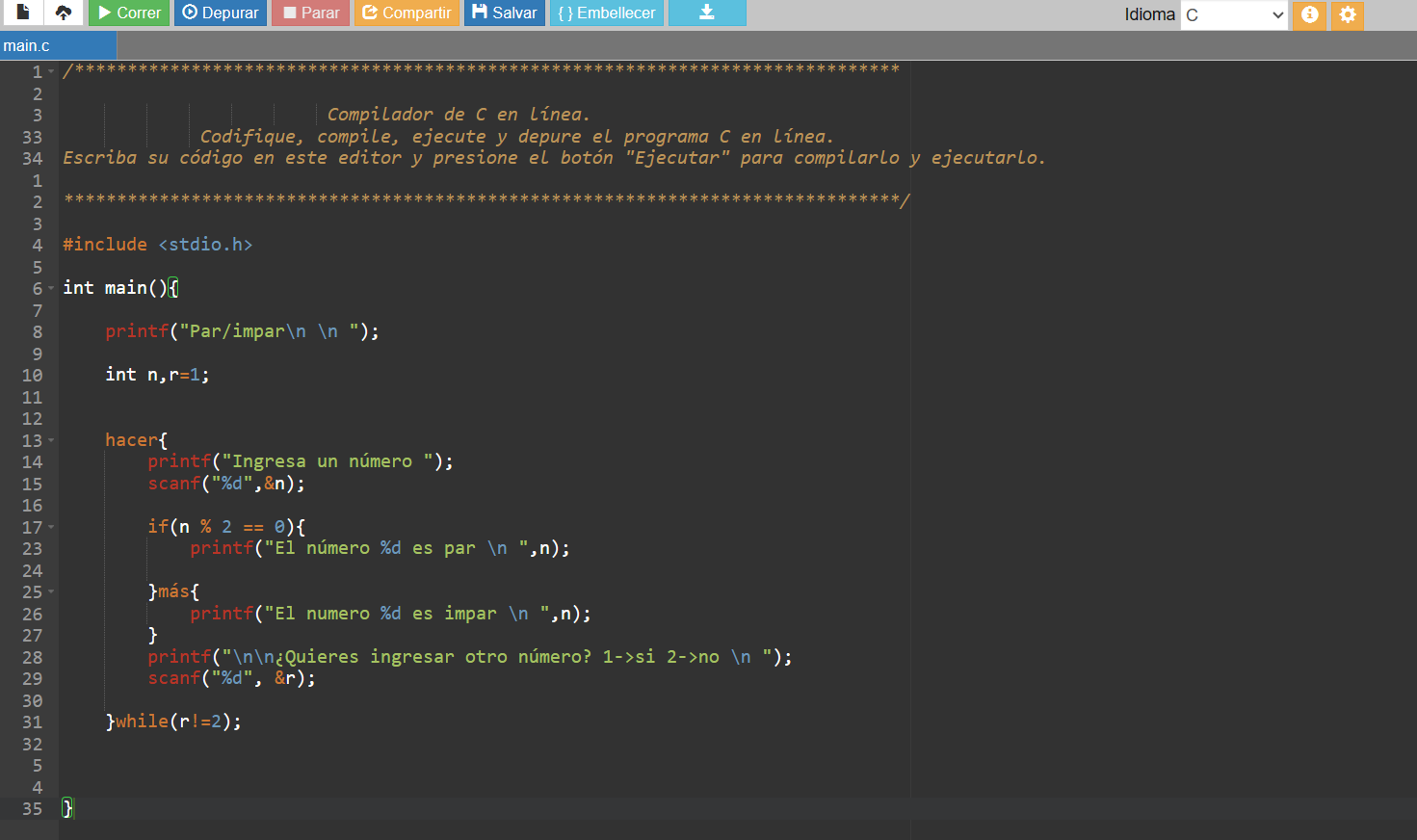
return 0;

}

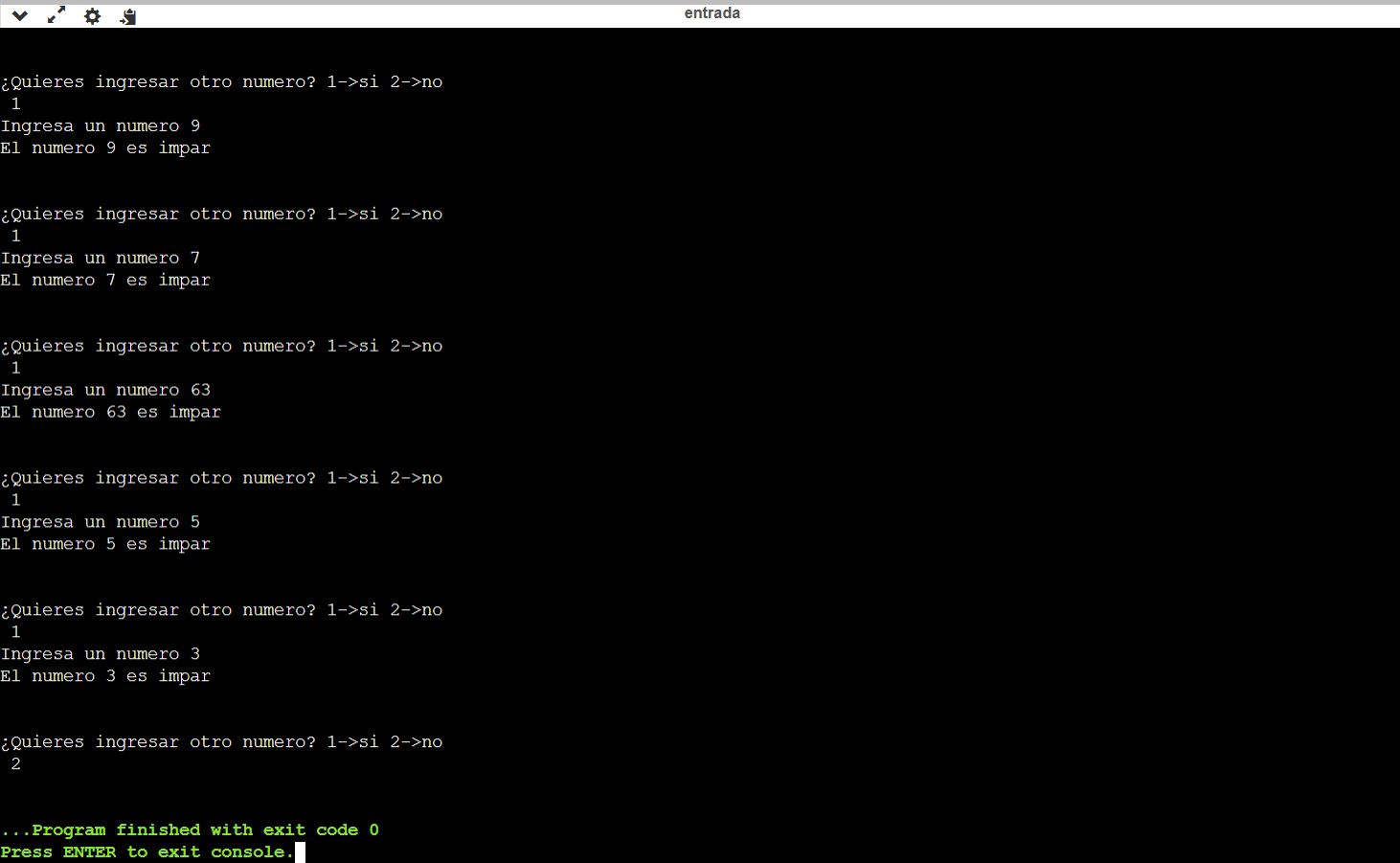
Calculadora Par e Impar

En esta calculadora lo ideal es que el usuario pueda colocar cualquier digito y este programa le arroje un mensaje de que, si ese número es par o impar, les dejare los ejemplos en el Shell del código ejecutado. Esta calculadora detectara cualquier número que arroje y saber si es par o impar.

Captura del código en DEV++:



Código ejecutado dando resultado como este número es Par



Código calculadora Par e Impar

#include <stdio.h>

int main(){

printf("Par/impar\n \n ");

int n,r=1;

do{

printf("Ingresa un numero ");

scanf("%d",&n);

if(n % 2 == 0){

printf("El numero %d es par \n ",n);

}else{

printf("El numero %d es impar \n ",n);

}

printf("\n\n¿Quieres ingresar otro numero? 1->si 2->no \n ");

scanf("%d", &r);

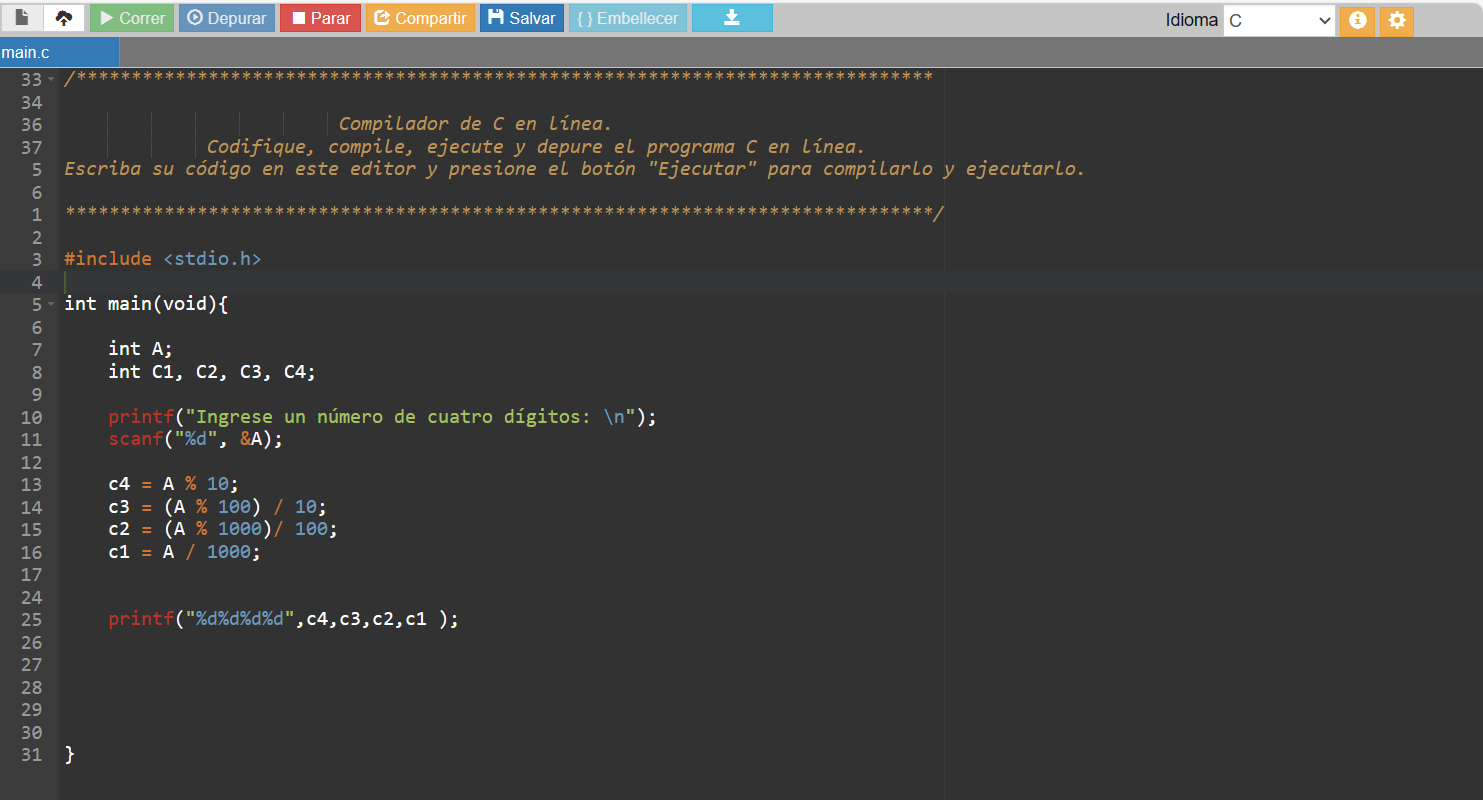
}while(r!=2);

}

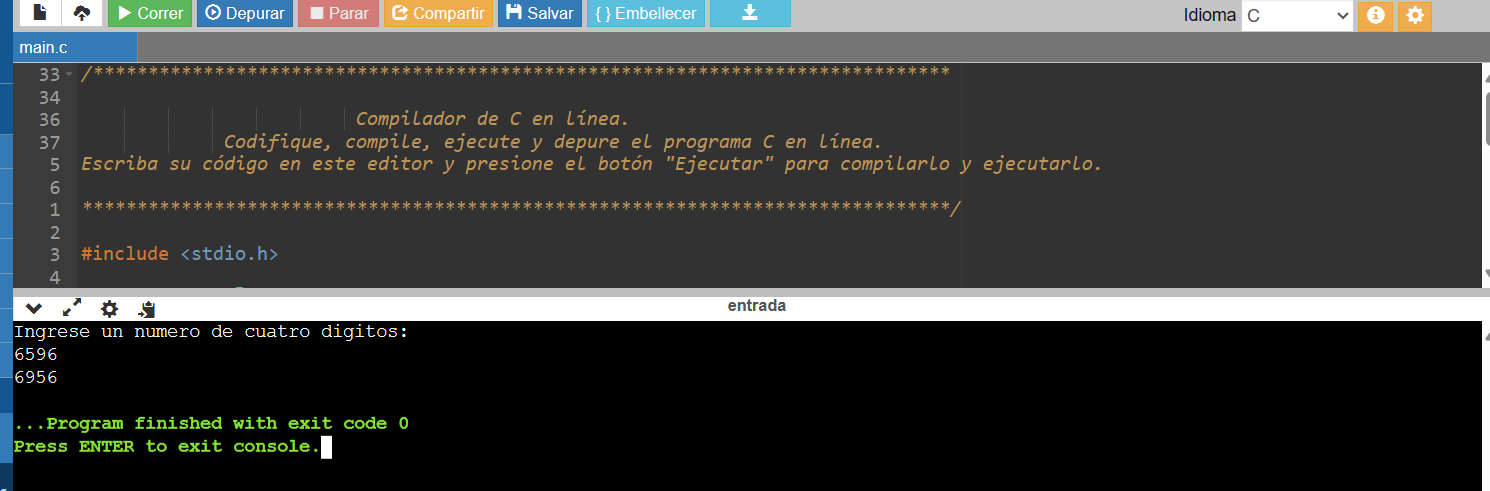
Calculadora Números al revés

En esta calculadora lo ideal es que el usuario pueda introducir un numero de 4 dígitos y que la calculadora tenga la capacidad de poder voltear esos números y poder darle otros valores, fue la calculadora más sencilla de crear ya que solo lleva un dato verdadero.

Captura del código en DEV++:



Código ejecutado volteando los dígitos añadidos:



Codigo de calculadora Numeros al reves:

#include <stdio.h>

int main(void){

int A;

int c1, c2, c3, c4;

printf("Ingrese un numero de cuatro digitos: \n");

scanf("%d", &A);

c4 = A % 10;

c3 = (A % 100) / 10;

c2 = (A % 1000)/ 100;

c1 = A / 1000;

printf("%d%d%d%d",c4,c3,c2,c1);

}

Conclusión

En esta actividad podemos llegar a la conclusión que con solo un algoritmo, código y un diagrama de flujo podemos llegar a programar una calculadora de manera rápida.

La importancia de saber usar el lenguaje C es muy importante ya que si lo dominas por completo podrás crear grandes proyectos con él, es un lenguaje muy extenso y entretenido de usar, puedes usarlo hasta para ti mismo. Creando programas que te ayuden en tus tareas de matemáticas, para crear proyectos para que puedas ofrecerlos a micro empresas y lo más importante es que te estarás familiarizando con otros lenguajes, podrás usarlos y entenderlos más fácil por ejemplo Python, C++ y C#, lo recomendable es no ir aprendiendo de varios lenguajes primero domina uno después el siguiente y asi sucesivamente, tendrás más oportunidades de trabajo, se abrirán más puertas a tu vida laboral y aprenderás mas cosas interesantes en este mundo de la programación.