Instituto Politécnico Nacional.

Escuela Superior de Cómputo

ESCOM

Profesor: Joel Omar Juárez Gambino

Alumno:

Torres Abonce Luis Miguel

Grupo:3CM3

Materia: Análisis y diseño de algoritmos

Tema: División de Gauss con Divide y Vencerás

Fecha: 02 de Noviembre 2021

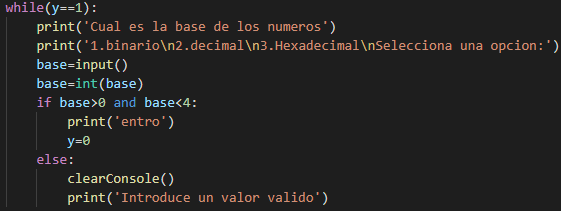
Introducción

En este programa se plantea el problema de resolver una multiplicación de dos números sin importar la longitud, también se proporcionan los datos de las bases en que se multiplicará, esta multiplicación se deberá resolver con el método de “Multiplicación de Gauss” además de utilizar recursividad o en su caso una pila de llamadas, al final se deberá a dar a conocer el numero resultante de la multiplicación en la base que se especifico al principio del programa.

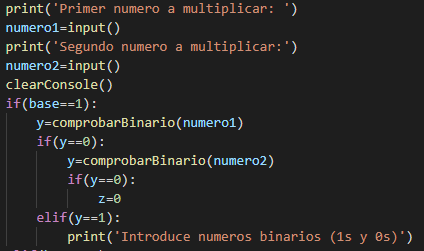
Decidí hacer mi programa en Python ya que nunca lo había utilizado y tenia ganas de programar en otro lenguaje que no sean los típicos como lo son c/c++ o java además que Python tiene bastantes librerías que se pueden aplicar a algunos problemas compleajos.

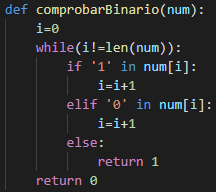
Desarrollo

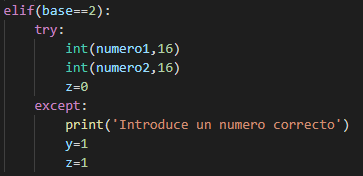
El programa lo trabaje con varias funciones además de la principal.

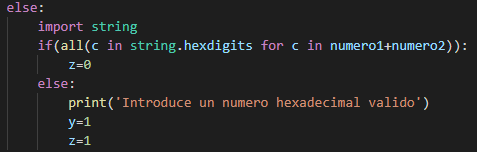


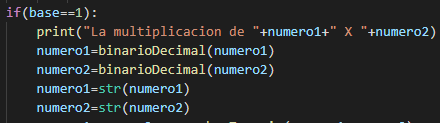
En esta primera parte del código lo único que se realiza es un menú que verifique que la respuesta este en 1 y 4 que son las opciones validas, esto para saber la base de los números

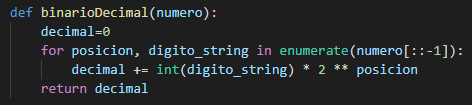
En esta parte se piden los 2 números a multiplicar y se verifica que si este numero es binario que realmente sea binario con la función comprobar binario, si esto es incorrecto se manda un mensaje que el numero es incorrecto y regresa al menú.

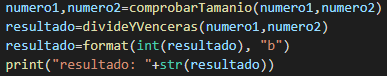
En esta función se verifica que sea un numero binario esto se consigue con un while que recorre todo el arreglo y unas sentencias if y elif que lo que hacen es verificar si hay un 1 o 0 en la cadena de caracteres cuando recorre todo regresa true o false dependiendo del resultado

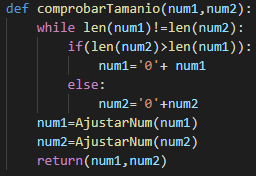
aquí si la base es decimal se ocupa un try except donde el programa intenta convertir la cadena a un entero si esta es correcta avanza sino puede y da error ejecuta el except donde se indica que es un numero incorrecto y regresa al menú

Para comprobar que sea hexadecimal se importa un string y se utiliza una condición donde con hxdigits es una cadena que contiene todo los numeros hexadecimales y regresa true o false dependiendo el caso si esto se cumple en las 2 cadenas sigue por el contrario imprime un mensaje donde se especifica que el numero no es correcto

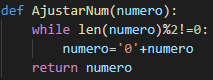
Si la base es decimal se convierte en decimal con la función binarioDecimal para trabajar el divide y venceras además se convierte a cadena el numero ya que lo regresa como int.

Se recibe una cadena con numero binario y se convierte usando la formula: num\_bin\*2(base)posicion

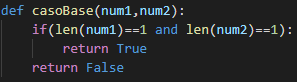
Se comprueba que el tamaño sea igual con la función comprobartamanio y se le asigna a los 2 numero lo que regresa, posteriormente al resultado se le iguala lo que regrese el divide y venceras, para finalmente convertirlo en binario con format y se imprime el resultado.



En esta función se comprobará que el tamaño de las 2 cadenas sea iguales esto se logra con un simple while y con una sentencia para saber en cual de los dos números se debe agrega el cero, una vez echo se mandan los dos números a la función ajustar numero para verificar que sean potencia de 2 y se regresan las 2 cadenas ya con el mismo tamaño y potencia de 2.

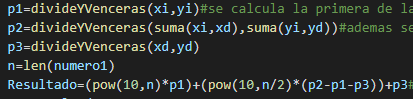
La primera es ajustar número esta recibe un string y le agrega ceros a la izquierda si el tamaño del string no es potencia de 2 y regresa el string ya con los ceros.

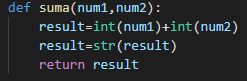
En la primera parte de divide y vencerás se verifica el caso base y si es true se regresa la multiplicación de los 2 números recibidos.

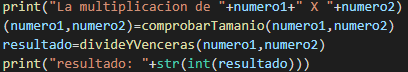
En el caso base se reciben los dos números y se verifica que el tamaño de los don números sea 1 y si es correcto se regresa true por lado contrario si es incorrecto se regresa false

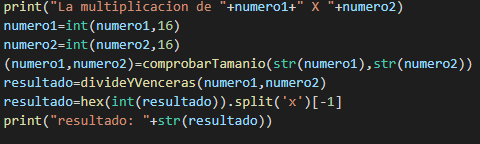
después se comprueba el tamaño de los 2 números para que estos sean iguales y se puedan dividir a la mitad, posteriormente se dividen los números a la mitad y se asigna a su parte izquierda y derecha.

El número que se recibe se divide en dos teniendo en cuenta su tamaño ya que siempre se podrá dividir igualmente y se regresa el número dividido.

En la última parte de divide y vencerás se calculan las 3 variables que se utilizan en el método de “Multiplicación de Gauss” las variables se obtienen a partir de utilizar recursividad con la función divide y vencerás por ultimo se calcula el resultado con todas las variables (p1,p2,p3) y se regresa este resultado que esta en decimal

Recibe 2 cadenas y multiplica su contenido que regresa igualmente convertida a cadena.

Para el caso donde los números son decimales no es necesario la conversión ya que divide y vencerás utiliza números decimales para el cálculo, lo demás es igual que si la base es binaria

Para el caso donde los números son hexadecimales se convierten a decimales y se realiza el mismo procedimiento que los 2 anteriores, la única diferencia es que al final el resultado está en decimal se convierte en hexadecimal y se imprime.

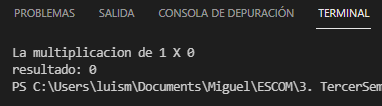
Pruebas.

Binario

Numero 1 = 1

Numero 2 = 0

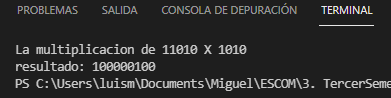
Resultado = 0



Numero 1 = 11010

Numero 2 = 1010

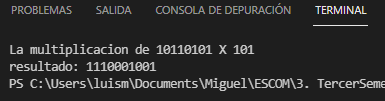
Resultado = 100000100



Numero 1 = 10110101

Numero 2 = 101

Resultado = 1110001001

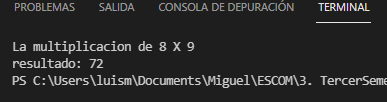


Decimal

Numero 1 = 8

Numero 2 = 9

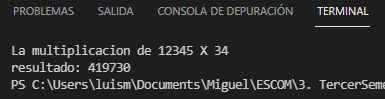
Resultado = 72



Numero 1 = 12345

Numero 2 = 34

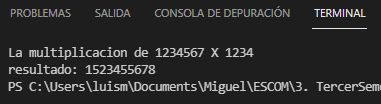
Resultado = 419730



Numero 1 = 1234567

Numero 2 = 1234

Resultado = 1523455678

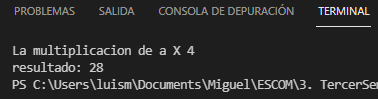


Hexadecimal

Numero 1 = A

Numero 2 = 4

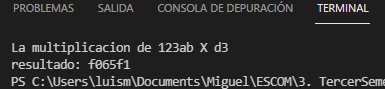
Resultado = 28



Numero 1 = 123AB

Numero 2 = D3

Resultado = f065f1



Numero 1 = 111222E

Numero 2 = 12CD

Resultado = 140f1f9ad6

