Instituto Politécnico Nacional

ESCOM

Ingeniería en Sistemas Computacionales

Fundamentos de Programación

Practica 03

Boleta: 2021630738

Torres Abonce Luis Miguel

Grupo: 1CV16

Recursividad

Objetivo de la práctica.

Resolver problemas Recursivos

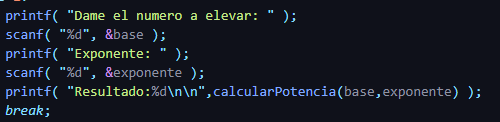
Introducción

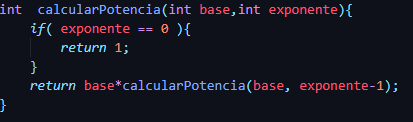
En esta práctica se desarrollarán los problemas propuestos por el profesor de manera recursiva, así como de manera iterada, los problemas eran los siguientes: calcular la potencia de un número, calcular la n suma de los primeros números, saber si un número es par o impar y calcular la multiplicación de 2 números.

Desarrollo.

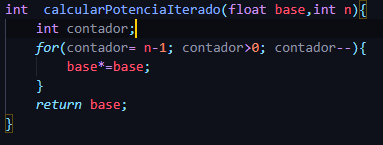
Primeramente, hice un menú para los 4 ejercicios que se asignaron, recursivamente, iterado, las torres de hanoi iterado y recursivo, posteriormente para cada uno de los 4 ejercicios les hice un menú a cada uno.

Para el primer problema era calcular la potencia de un número, como primer paso mande imprimir a pantalla al usuario el número y el exponente que quería elevar dicho número, posteriormente lo escanee con la función scanf, y mande imprimir lo que me va a devolver la función calcular potencia recursivo e iterado, para la versión recursiva primeramente puso el paso base que si el exponente es igual a 0 regrese un 1, pero si no es así regresa la multiplicación de la base por la misma función pero se le resta uno al exponente esto se repite hasta que sea 0 y regrese 1 después regresa la llamada y hace la sumatoria de todos los resultados.

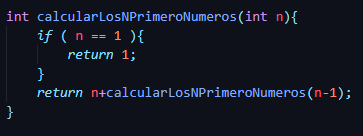




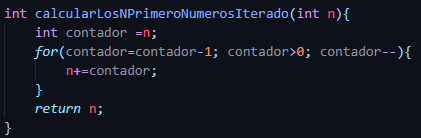
Para la versión iterada utilice un contador dentro de un ciclo for que empieza en exponente -1, contador termina cuando sea mayor que 0 y se le resta 1, dentro del for hace el proceso base = base \*base y cuando termina el ciclo se regresa la base final que sería el resultado.

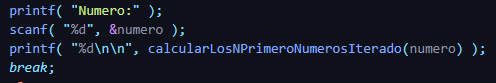


EL segundo problema para calcular los primeros N primos de un número, inicialmente se le pide al usuario que introduzca el numero que quieres sumar sus primeros números después se manda llamar para la recursividad o iterado según sea el caso, para la forma de recursividad se pone como paso base si n==1 entonces regresamos 1 a la función pero si no es uno entonces a el numero que nos dio el usuario (n) se le suma lo que regrese la función n-1 hasta que esta sea 1 regrese la llamada y sume todos los resultados.

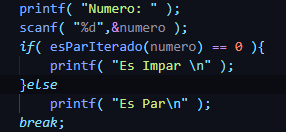


Para la forma iterada se manda el numero dado por el usuario (n) y se declara una variable contadora y es igualada a “n”, después se inicia un ciclo for donde la variable contador empieza en contador -1, termina cuando contador sea mayor que 0, y al contador se le resta 1 dentro de este ciclo a la variable n se la suma el contador para así llevar la cuenta de la suma de los n primeros numero de n y al final se retorna a la función n después del ciclo.

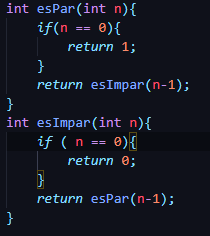




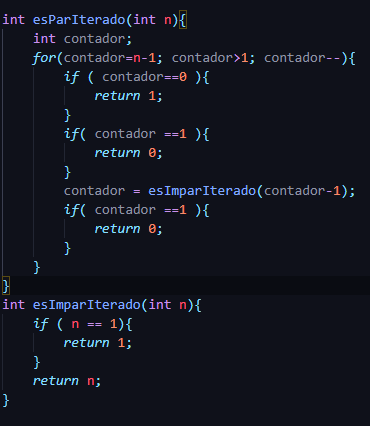
Para el tercer problema se le pide al usuario el numero que quiere saber si es par o es impar y se guarda en la dirección de la variable numero después se utiliza una sentencia de control if para saber si lo que devuelve la fduncion es impar es igual a 0 entonces se imprime en pantalla es impar, sino es par.



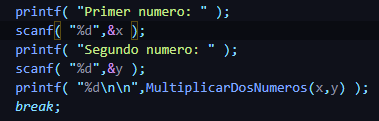
Dentro de la función esPar en la forma recursiva se hace lo siguiente se tiene dos funciones que se llaman mutuamente hasta que el paso base se cumpla en alguna de las 2, en es par se pone una sentencia if si el numero (n) es 0 entonces se regresa 1 a la función pero si no se regresa lo que te de la función es impar -1 para que se le vaya restando 1 unidad y llegue al paso base y se hacel lo mismo en la función es impar solo que aquí si “n” es igual a 0 se regresa 0 y para diferenciar si es par o impar.



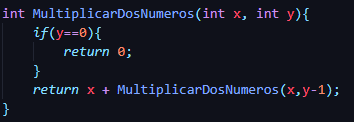
En la manera iterada se declara un contador y se inicia un ciclo for donde contador se iguala a n-1, contador termina cuando sea mayor que 1 y se le resta 1 en cada ciclo, dentro de este ciclo tenemos 2 sentencias if donde si contador es igual a 0 entonces se regresa 1 y sabremos que es par en la otra sentencia if donde si contador es iguala a 1 regresa cero y sabemos que es impar, después se iguala el contador a la función es impar y se le manda el contador -1 para llevar un control sobre este y después se repite la sentencia if donde contador es igual a 1 regresa 0, en la función es impar tenemos una sentencia donde si n es igual a 1 regresamos 1 pero si no se cumple esto entonces solo regresamos lo que le llego a la función.



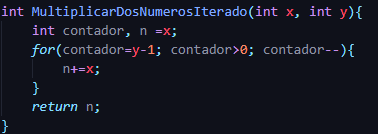
En el último problema donde se debe de realizar la multiplicación de 2 números para la versión de recursividad se le pide que el usuario de 2 números y se escanea y guarda en la dirección “x” y “y” y después se imprime lo que devuelva la función multiplicar dos números y se mandan los datos mencionados.



Para la versión de recursividad al principio va el paso base donde si y es igual a 0 se regresa 0 pero si no regresamos la suma de la variable x con la misma función multiplicar dos numero y se le mandan los datos x, y-1 para cuando regrese la llamada se tenga la cuenta de cuantas veces se debe de sumar la x a si misma.



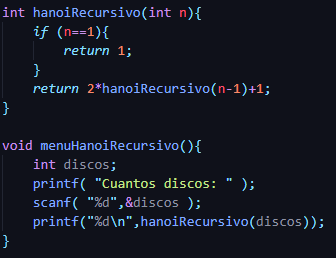
Para la versión iterada se declaran 2 variables el contador y n que se iguala a x después se incicia un ciclo donde el contador se iguala a y-, el ciclo terminara cuando contador sea mayor que 0 y al contador se le resta 1 con cada ciclo que da dentro del ciclo se lleva a acabo de n mas x esto se sumara hasata que contador sea 0 y finalmente se regresara n fuera del ciclo que seria el resultado final.



Para el problema de torres de Hanói al igual que los anteriores se realizo 1 menú para cada una de las maneras de realizar el problema tanto de manera iterada, así como de manera recursiva donde permite al usuario elegir como quiere que se realice.

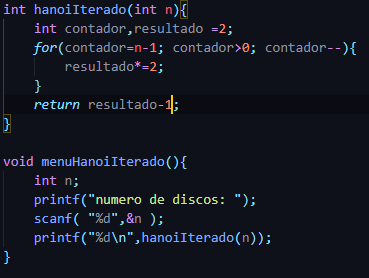
Para la forma recursiva se realizó:

Primeramente se puso un paso base donde si n que es el numero de discos otrogada por el usuario es igual a 1 entonces se regresa 1, por lo contario se regresa la función hanoi mandole el dato n-1, esta función multiplicando por 2 y sumándole 1 al final.



Para la forma iterada se realizó:

Se declararon las variables contador y resultado que se iguala a 2, con un ciclo for donde el contador empezara en n-1, terminara cuando contador sea mayor que 0 y al contador se le resta 1. Y fuera de este ciclo se regresa el resultado -1.



Conclusiones.

La practica estuvo sencilla y realmente comprendí un poco mas para que es la recursividad y lo veo como otra forma de resolver un problema planteado, lo único es que hubo un momento donde me quede atorado en las librerías ya que no regrese a compilar la práctica con terminación en c, sino que estaba compilando la librería y no se actualizaba el documento de la practica y al ejecutarlo no hacía ningún cambio hasta que busque en internet alguna solución y leí que las librería no se deben de compilar sino el documento principal.