	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor:	TISTA GARCÍA EDGAR
Asignatura:	ESTRUCTURA DE DATOS Y ALGORITMOS I
Grupo:	1
No de Práctica(s):	PRÁCTICA #12. RECURSIVIDAD
Integrante(s):	CARRILLO CERVANTES IVETTE ALEJANDRA JUAREZ JUAREZ MARIA JOSE
No. de Equipo de cómputo empleado:	TRABAJO EN CASA
No. de Lista o Brigada:	Brigada 2
Semestre:	2021 - 2
Fecha de entrega:	02 AGOSTO 2021
Observaciones:	

CALIFICACIÓN: _____

PRÁCTICA #12 – RECURSIVIDAD

OBJETIVO: APLICAR EL CONCEPTO DE RECURSIVIDAD PARA LA SOLUCIÓN DE PROBLEMAS

ACTIVIDADES PARA EL DESARROLLO DE LA PRÁCTICA

1.- Explica en tus palabras las principales ventajas y desventajas de la recursividad

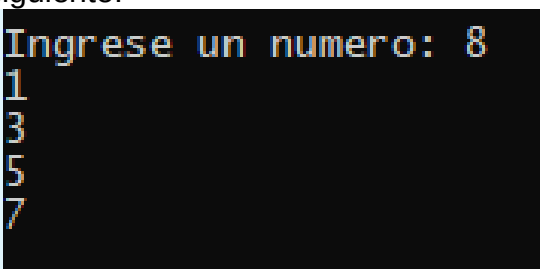
Se puede utilizar cuando no existe una manera de resolver el problema de manera simple con una solución iterativa, puede reducir la complejidad temporal de ciertos algoritmos.
Puede hacer más sencilla la depuración del código, y el análisis de los algoritmos.
En ocasiones puede ser más lenta y ocupar un mayor espacio en la pila.

2.- Codifica, compila y ejecuta el siguiente programa e indica qué hace

En el caso de este programa, tuvimos que añadir la función principal, donde se mandará llamar la función recursiva. Así como, la opción de que el usuario eligiera el número a considerar para la ejecución de las operaciones. Este programa tiene como función principal, dado un número “n” ingresado por el usuario, imprimir los números impares que van de cero hasta “n”; para ello, dentro de la función cat (la cual recibe el número ingresado por el usuario como parámetro), se realizó dos condicionales:

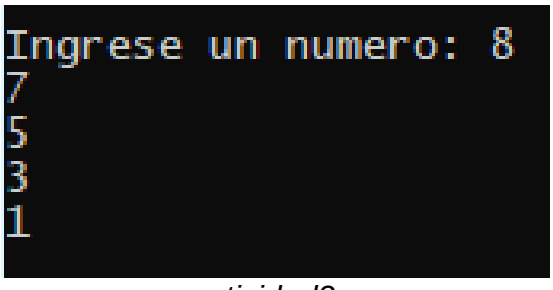
- Si el módulo del número ingresado es 0, entonces la variable donde está almacenado este número “n”, disminuirá 1 puesto que el número ingresado será un número par y se requiere que sea un número impar.
- Si el número es mayor que 0, se llamará a la misma función con el número “n-2” como parámetro y se imprimirá

La salida del programa es la siguiente:



actividad2.c

Invierte el orden de las instrucciones del segundo if, explica el porqué de los resultados obtenidos
Al invertir las instrucciones del código en el segundo if, ahora en vez de imprimirse de menor a mayor, se imprime de mayor a menor, esto se debe a que primero se imprime el número y después se llama a la función recursiva.



actividad2.c

3.- Codifica, compila y ejecuta el siguiente programa e indica qué hace

Este programa tiene como función principal sumar los elementos de una lista dado un índice, este índice será el “tope” de los elementos que se quieren sumar; para ello, se creo una función cuyos parámetros son una lista previamente inicializada con elementos los cuales son múltiplos de dos

```
12   lista1 = [2, 4, 6, 8, 10, 12]
```

y un número “n”, el cual se modificó para que el usuario ingrese el índice, que marca la posición de la lista hasta donde se quiere sumar. Dentro de esta función se hizo la siguiente condición:

- Si el número ingresado es 1, la función retornará solo el primer elemento de la lista
- De lo contrario, retornará la suma del primer índice más el resultado que retorne la función al volverla a llamar cuyos parámetros ahora son, el siguiente de la lista y n-1

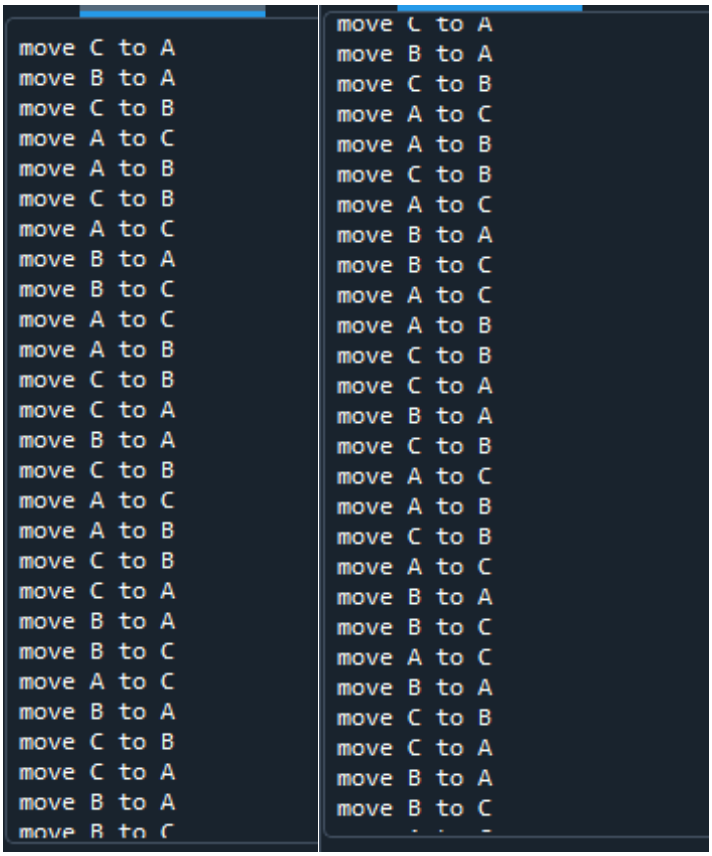
La salida del programa es la siguiente:

```
C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Práctica G1 P12 V1
>actividad3.py
Ingrese n: 4
20

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Práctica G1 P12 V1
>actividad3.py

Ingrese n: 1
2

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Práctica G1 P12 V1
>actividad3.py

Ingrese n: 2
6

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Práctica G1 P12 V1
>actividad3.py

Ingrese n: 3
12

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Práctica G1 P12 V1
>actividad3.py

Ingrese n: 4
20

C:\Users\aleja\OneDrive\Escritorio\FACULTAD\2021-2\EDA\Prácticas\Práctica G1 P12 V1
>
```

actividad3.py

4.- Codifica y compila el siguiente programa, explica cómo se resuelve el problema de las Torres de Hanoi con este programa

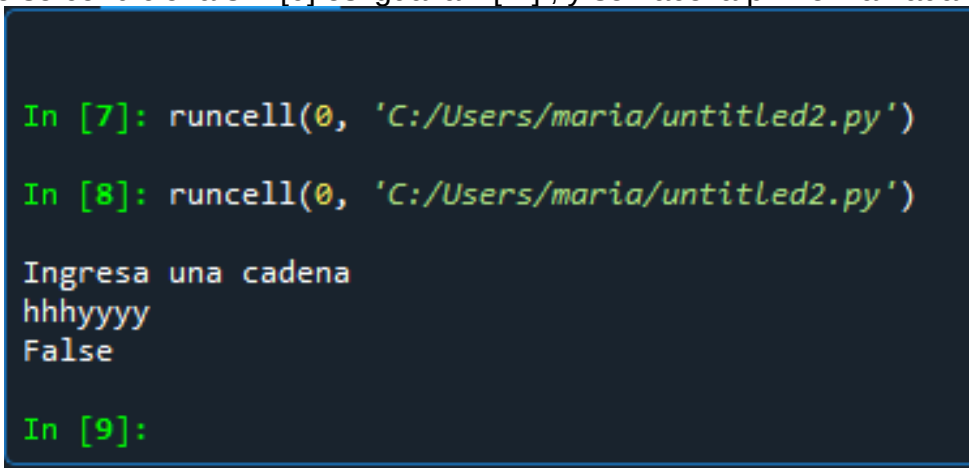
Se reciben 4 parámetros, tres de tipo char y uno de tipo entero, seguido de esto se entra a una condicional, donde sí el número n es igual a 1 se imprimirá en pantalla “move “x “to” y, x y ‘y’ son los caracteres proporcionados anteriormente como parámetros. En caso de que esa condición no se cumpla, se entrará en un else donde está contenida la primera llamada recursiva, pasando parámetros distintos a los iniciales. En este caso en lugar de n, será n-1, y los demás parámetros permanecen como inicialmente estaban definidos, después se imprimirá el mismo enunciado que en la primera condicional, pero, ya alterado por el n-1.
Luego de imprimir se vuelve a hacer una llamada recursiva a la función, donde se usan los mismos parámetros que en la llamada anterior.
Esto logra que se sigan las reglas de la torre de hanoi, sobre el hecho de no poder colocar una pieza más grande sobre una más pequeña, solo se puede mover a la vez un disco. La salida del programa es la siguiente:



actividad4.py

5.- Indica qué hace el siguiente programa

Este programa pide al usuario una cadena de caracteres, que será el parámetro de la función mystery, se crea una variable N para la que se utiliza la funcion split sobre la cadena S que devuelve una lista de palabras, se utiliza basicamente para dividir un string en cadenas mas pequeñas.
Mas adelante en la misma variable N, se utiliza la funcion join que sirve para convertir una lista en una cadena formada por los elementos de la lista separados por comas.
Se entra en una condicional donde se evalua si el tamaño de N es igual a 1 o igual a 0, si se cumple alguno de estos casos se retorna el valor booleano True, de no ser asi se entra en lel siguiente if anidado, donde se condiciona si N[0] es igual a N[-1] , y se hace la primer llamada recursiva.



Actividad5.py

Conclusiones:

Carrillo Cervantes Ivette Alejandra

Se cumplieron con los objetivos de esta práctica, ya que se aplicaron funciones recursivas para solucionar ciertos problemas dados, la diferencia entre una función recursiva y una función iterativa es bastante notoria, y a principio, personalmente un poco difícil de interpretar, pero poco a poco se fue entendiendo mejor. Considero que las actividades realizadas si ayudaron al concepto visto en clase.

Se cumplió con el 100% de las actividades propuestas para el desarrollo de la práctica, en el ejercicio 5, se nos complicó un poco al hacer la interpretación de los datos; sin embargo, checándolo más a detalle, nos pudimos dar cuenta que la función de este programa fue saber si una cadena tiene el mismo número y letra en todos sus índices.

Juárez Juárez María José

Esta práctica fue útil para comprender la diferencia entre utilizar algoritmos iterativos y algoritmos recursivos, así como para entender los casos en los que es mejor utilizar unos u otros y el porqué de esto, las ventajas y desventajas de utilizarlos y la manera en que se analizan los algoritmos recursivos, un poco más compleja y distinta de lo que vemos comúnmente en los iterativos.