

PRACTICA Nº 1

(ESTRUCTURA CONDICIONALES, REPETITIVAS, LISTAS, POO)

1. Realizar un programa que permita pedir el día, mes y año de una fecha correcta y mostrar la fecha del día siguiente. suponer que todos los meses tienen 30 días.
2. Realizar un programa que determine el menor de entre grupo de números enteros.
3. Realizar un programa que reciba 3 números enteros y luego
 - a) Muestre el número intermedio del conjunto de los 3 números.
 - b) Presente los 3 números en forma descendente.
4. Realizar un programa que pida una cadena y un carácter por teclado (valida que sea un carácter) y muestra cuantas veces aparece el carácter en la cadena.
5. Realizar un programa que permita ingresar por teclado una cadena la misma que debe ser invertida, mostrar por pantalla ambas cadenas.
6. Realizar un programa que permita leer un número y mostrar su cuadrado, repetir el proceso hasta que se introduzca un número negativo.
7. Realizar la operación de potenciación (a^b), de dos valores enteros positivos, con multiplicaciones.
8. Escribir un programa que sume las cifras de un número entero positivo.
Ejemplo: $962 \ 9 + 6 + 2 = 17$
 $1 + 7 = 8$
El resultado es 8
9. Realizar un programa que invierta un número introducido por teclado. Debe solicitar un valor entero y mostrar el mismo número con sus cifras invertidas. Si el número es negativo debe seguir siéndolo.
 $-12345 \rightarrow -54321$
 $54300 \rightarrow 00543$
10. Para un conjunto de n datos reales se desea determinar el mayor de los datos negativos y cuantas veces aparece. Por ejemplo para n=6, datos: 4,-5,6,-2,5,-3.
11. Escribir un programa que muestre la figura de caracteres siguiente. El valor del carácter máximo mostrado se le debe pedir al usuario. Los únicos valores válidos son del 'a' a la 'i'.

```
      a
    a b a
  a b c b a
a b c d c b a
  a b c b a
    a b a
      a
```

12. El programa debe dar un mensaje de error cuando el número N no está en los límites indicados y solicitar nuevamente el valor N al usuario.

Ejemplo para N=4

```

-   +   +   +
*   -   +   +
*   *   -   +
*   *   *   -

```

13. Los pacientes con síntomas de una cierta enfermedad son ingresados en el hospital si tienen un valor superior a 0.6 en la medición de un determinado índice, y son operados si el valor es superior a 0.9. Escribir un programa que lea desde teclado el número de pacientes seguido de la edad y el índice de cada paciente, y calcule la edad media de los pacientes analizados así como la edad media de los ingresados y la edad media de los operados.

14. Realizar un programa que pida números y los guarda en una lista, cuando el usuario meta un 0 ya dejaremos de insertar. Por último, muestra los números ordenados de menor a mayor.

15. Realizar un programa que tome un string y devuelva un diccionario cuyas llaves son las letras del string y sus respectivos valores son la cantidad de veces que ellas aparecen en el string. Por Ejemplo, contador('telecomunicaciones') {'t': 1, 'e': 3, 'l': 1, ...}.

16. Realizar un programa que intercambie claves y valores en un diccionario.

17. Realizar un programa aplicando la POO de la clase Artículo Científico:

ArticuloCientifico
Titulo:String Autor:String PalabrasClaves:String Publicación:String Año:int Resumen: String
__init__() Mostrar_articulos()

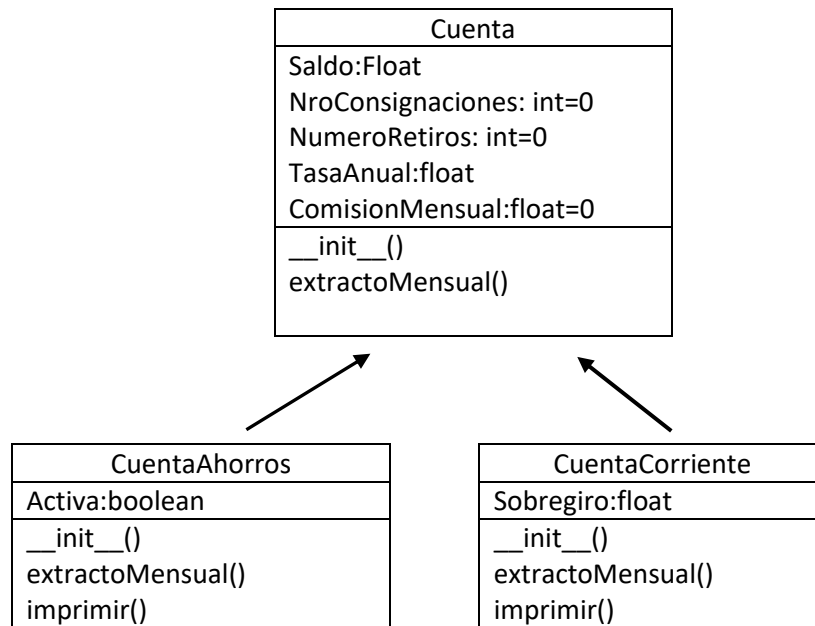
a) Instanciar N objetos y almacenarlos en una lista.

18. Realizar un programa aplicando la POO de la clase Agenda Telefonica:

ArticuloCientifico
Nombre:String Apellido:String NroCelular:String Correo:String
__init__() Adicionar_Contacto() Mostrar_Contactos() Buscar_Contacto

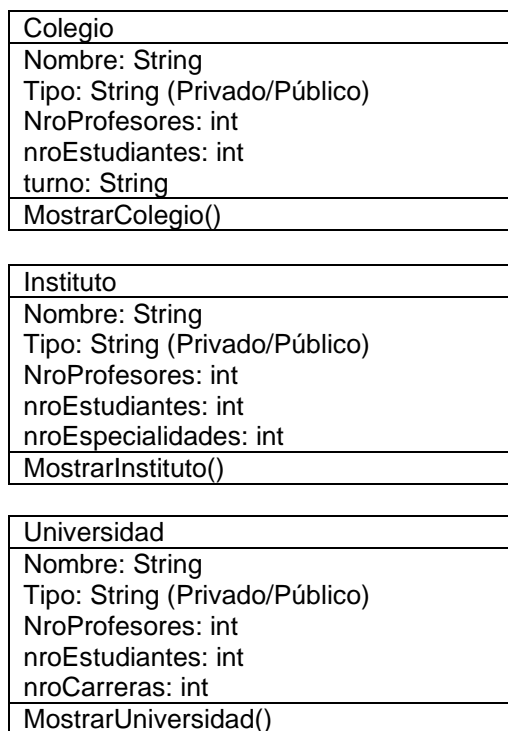
a) Instanciar N objetos y almacenarlos en una lista.

19. Desarrollar un programa aplicando la POO de la clase padre Cuenta y las clases hijas Cuenta_Corriente y Cuenta_Ahorros



- Instanciar 3 objetos
- Aplicar el principio de Herencia

20. Sean Colegio, Instituto Técnico y Universidad, entre instituciones educativas públicas y privadas.



- a) Instanciar 3 objetos de cada una de las clases.
- b) Mostrar un colegio, un instituto y una universidad.
- c) Aplicar el principio de Herencia
- d) Aplicar encapsulamiento en los atributos y métodos.