

UNAC FIME

CÁLCULO NUMÉRICO COMPUTACIONAL

TAREA 1

PROGRAMACIÓN PYTHON - 30 PROBLEMAS

Fecha de Entrega : 14 de Septiembre de 2025

Instrucciones

Resuelva los siguientes 30 problemas de programación en Python. Los problemas están divididos en tres niveles: básico, y intermedio.

- Para cada problema entregue:
 1. **Incluya comentarios en su código y asegúrese de que cada programa funcione correctamente.**
 2. Código fuente,
 3. Un archivo de prueba o ejemplo de entrada/salida,
 4. Breve explicación (máx. 10 líneas) de la solución y decisiones de diseño.
- Es obligatorio emplear buenas prácticas: nombres claros, funciones pequeñas, documentación mínima (docstrings) y control de errores.

A continuación se listan 30 problemas divididos por nivel. Cada problema indica **objetivo**, **entregable** y **pistas/hints**.

1 Problemas Básicos (10 problemas)

1. **Hello World Modificado**
Escriba un programa que solicite al usuario su nombre y muestre un mensaje de saludo personalizado.
2. **Calculadora Básica**
Cree una calculadora que realice las cuatro operaciones básicas (suma, resta, multiplicación, división) con dos números ingresados por el usuario.
3. **Par o Impar**
Programa que determine si un número ingresado por el usuario es par o impar.
4. **Factorial de un Número**
Calcule el factorial de un número usando un bucle `for`.
5. **Suma de Números**
Sume todos los números desde 1 hasta un número N ingresado por el usuario.
6. **Lista de Cuadrados**
Genere una lista con los cuadrados de los primeros 10 números naturales.
7. **Contador de Vocales**
Cuenta la cantidad de vocales en una cadena de texto ingresada por el usuario.
8. **Conversión de Temperatura**
Convierta temperaturas entre Celsius y Fahrenheit según la elección del usuario.
9. **Mayor de Tres Números**
Encuentre el mayor de tres números ingresados por el usuario.
10. **Generador de Tablas de Multiplicar**
Genere la tabla de multiplicar de un número dado hasta el 10.

2 Problemas Intermedios (20 problemas)

Problema 1

Escribir un programa en Python que permita introducir dos enteros, A y B, por teclado, y calcule y muestre en pantalla su suma y su producto, con un texto explicativo. Además, el programa debe comparar los valores A y B y debe informar por pantalla del mayor de ellos.

Problema 2

Escribir un programa en Python de calculadora mínima, que sea capaz de sumar, restar, multiplicar y dividir (incluir resto de la división también) dos números introducidos por teclado. Utilice el siguiente menú secuencial:

1. Introduzca dos números enteros.
2. Introduzca el tipo de operación (mediante símbolo carácter o entero). (Considerar la posibilidad que la operación introducida no sea válida, y hay que descartar que se divide por cero.)

Problema 3

Escribir un programa en Python que, dados tres números introducidos por teclado, los represente en la pantalla ordenados de mayor a menor.

Problema 4

Escribir un programa en Python que, dados tres ángulos introducidos por teclado, informe de si forman triángulo y si el triángulo formado es rectángulo, acutángulo o obtusángulo.

Problema 5

Escribir un programa en Python que, dados una base y un exponente (entero positivo o cero) introducidos por teclado, devuelva el resultado de realizar la operación de elevar la base al exponente. No se puede utilizar la función exponencial de Python.

Problema 6

El factorial de un número natural es igual al producto de todos los naturales menores o iguales que él. El factorial de 0 es 1. Escribir un programa en Python que muestre el factorial de un número natural introducido por teclado.

Problema 7

Escribir un programa en Python que calcule el máximo como un divisor de dos números naturales dados utilizando el algoritmo de Euclides.

Problema 8

La sucesión de Fibonacci se obtiene a partir de sus dos primeros términos, ambos igual a 1, de forma que cada término posterior es la suma de los dos términos anteriores.

1	1	1	2	3	5	8	13	. . .
2			1+1	1+2	2+3	3+5	5+8	

Escribir un programa en Python que muestre por pantalla el número de la sucesión de Fibonacci que ocupa la posición indicada por teclado.

Problema 9

Un número natural es perfecto si es igual a la suma de sus divisores propios. Escribir un programa en Python que, dado un número natural introducido por teclado, informe de si es un número perfecto o no.

Problema 10

Escribir un programa en Python que, dado un numero natural introducido por teclado, muestre por pantalla todos los números perfectos menores que el.

Problema 11

Un número es perfecto si es igual a la suma de n-primeros números naturales.

Sum	Number of Correct Answers
1	5
1+2	3
1+2+3=6	1
...	5

Escribir un programa en Python que, dado un número natural introducido por teclado, informe de si es un número triangular o no.

Problema 12

Escribir un programa en Python que, dadas dos fechas introducidas por teclado, indique el orden cronológico de las fechas. Utilizar vectores para almacenar las fechas.

Problema 13

Escribir un programa en Python que, dadas las temperaturas de un mes introducidas por teclado, calcule y muestre la temperatura máxima, la mínima y la media. Utilizar un vector para almacenar las temperaturas y una constante para el número de días del mes.

Problema 14

Escribir un programa en Python que, dadas las temperaturas de un mes introducidas por teclado, compruebe si otra temperatura, introducida también por teclado, se ha producido en el mes o no. Utilizar un vector para almacenar las temperaturas y una constante para el número de días del mes.

Problema 15

Escribir un programa en Python que, dadas las notas de un alumno en cuatro cursos compuestos por cinco asignaturas cada uno, calcule y muestre por pantalla:

- la media global del alumno
- la media de cada curso almacenada en un vector
- la media de cada asignatura almacenada en un vector

Las notas se introducirán por teclado y deberán almacenarse en una matriz. Para que el programa sea más general utilizar constantes para el número de cursos y asignaturas.

Problema 16

Dados dos números naturales $n \geq m \geq 0$ se define el número combinatorio n sobre m como

$$C_{n,m} = \binom{n}{m} = \frac{n!}{m!(n-m)!} \quad (1)$$

Escribir una función en Python que calcule el factorial de un número. Utilizar esta función para calcular el número combinatorio de dos números introducidos por teclado.

Problema 17

Escribir un programa en Python que calcule el Máximo Común Divisor y el Mínimo Común Múltiplo de dos números naturales introducidos por pantalla. Con este fin escribir:

- una función que calcule el Máximo Común Divisor de dos números naturales con el algoritmo de Euclides
- una función que calcule el Mínimo Común Múltiplo de dos números naturales utilizando la función anterior sabiendo que:

$$A * B = \text{mcd}(A, B) * \text{mcm}(A, B) \quad (2)$$

Problema 18

Realizar un programa en Python que representando las fracciones como vectores de 2 componentes enteras (numerador y denominador), implemente una serie de operaciones sobre estas. Debe ser capaz de sumar, restar, multiplicar y dividir dos fracciones y el resultado debe mostrarse como fracción irreducible.

Para dicho programa se utilizarán tanto funciones como subrutinas (al menos una por operación). El programa principal contendrá un menú en el que se permita elegir entre todas las operaciones posibles.

Problema 19

Realizar un programa en Python que implemente el juego del ahorcado. Para ello realizar las siguientes subrutinas:

- Subrutina que recibe como argumento el número de errores y dibuja el muñeco del ahorcado de la siguiente forma:

```

1 error = 1   error = 2   error = 3   error = 4   error = 5   error = 6
2   --         ---         ---         ---         ---         ---
3           |           |           |           |           |
4           0           0           0           0           0
5           /           /           /           /           /
6 error = 7   error = 8   error = 9   error = 10
7   ---         ---         ---         ---
8   |           |           |           |
9   0           0           0           0
10  /|\         /|\         /|\         /|\
11  |           |           |           |
12  /           /           /           /

```

- Subrutina que lee la palabra que escribe el primer jugador, la guarda en un vector de 20 caracteres, contabiliza el número de letras de la palabra y guarda en el resto del vector espacios en blanco. Además deberá ocultar la palabra al segundo jugador utilizando saltos de línea con la instrucción `print*, " "`.
- Subrutina que comprueba si la letra elegida por el segundo jugador pertenece o no a la palabra. En caso afirmativo debe mostrar en qué posición o posiciones aparece la letra utilizando otro vector de 20 caracteres que vaya mostrando los aciertos y los espacios de las letras que faltan por adivinar.

Ejemplo: Si la palabra es **CASA** y la letra elegida es **A** tendremos dos vectores para realizar el resto del programa

Palabra	C	A	S	A															
Aciertos	*	A	*	A															

se pueden implementar mas subrutinas/funciones o incluir mas funcionalidades en las subrutinas anteriores.

Problema 20

Realizar un programa en código Python que represente a los números complejos z como vectores de dos componentes ($a, b \in \mathbb{R}$) de la forma $z = a + bi$, (donde **a** es la parte real y **b**, la parte imaginaria). Implemente una serie de operaciones sobre estos, dichas operaciones serán:

1. Leer un número complejo.
2. Escribir un número complejo.
3. Multiplicar un número complejo por un número escalar.
4. Hallar el conjugado de un número complejo.
5. Sumar dos números complejos.
6. Restar dos números complejos.
7. Multiplicar dos números complejos.
8. Dividir dos números complejos.
9. Realizar el opuesto de un número complejo.
10. Realizar el inverso de un número complejo.

Nota 1: Para dicho programa se utilizarán subrutinas (Se realizará como mínimo una por operación).

- (a) Multiplicar un número complejo por un número escalar.

$$k * (a + bi) = k * a + k * bi$$

- (b) Hallar el conjugado de un número complejo. Conjugado

$$(a + bi) = (a - bi)$$

- (c) Suma de dos números complejos.

$$(a + bi) + (c + di) = (a + c) + (b + d)i$$

(d) Diferencia de dos números complejos.

$$(a + bi) - (c + di) = (a - c) + (b - d)i$$

(e) Producto de dos números complejos.

$$(a + bi) * (c + di) = (a * c - b * d) + (a * d + b * c)i$$

(f) División de dos números complejos.

$$(a + bi)/(c + di) = \frac{(a + bi) * (c - di)}{(c^2 + d^2)}$$

Utilizar obligatoriamente para este apartado los subprogramas ya implementados en los apartados: (e) la multiplicación de dos números complejos, (b) el conjugado y (a) la multiplicación de un escalar por un número complejo.

(g) Opuesto de un número complejo. Opuesto

$$(a + bi) = (-a - bi)$$

Utilizar para este apartado el subprograma del apartado (a)

(h) Inverso de un número complejo. Inverso

$$(a + bi) = 1/(a + bi)$$

Utilizar para este apartado el subprograma del apartado (f)

Nota 2: Se deberá realizar un programa principal con un menú en el que se permita elegir entre todas las operaciones posibles.

Problema 21: Algoritmo de multiplicación rusa

Consiste en:

1. Escribir los números (A y B) que se desea multiplicar en la parte superior de sendas columnas.
2. Dividir A entre 2, sucesivamente, ignorando el resto, hasta llegar a la unidad. Escribir los resultados en la columna A.
3. Multiplicar B por 2 tantas veces como veces se ha dividido A entre 2. Escribir los resultados sucesivos en la columna B.
4. Sumar todos los números de la columna B que estén al lado de un número impar de la columna A. Éste es el resultado.

Ejemplo: 27×82

A	B	Sumandos
27	82	82
13	164	164
6	328	
3	656	656
1	1312	1312
Resultado: 27 x 82 = 2214		

Código de Ejemplo

```
1 def saludo_personalizado():
2     nombre = input("Cual es tu nombre? ")
3     print(f"Hola {nombre}! Bienvenido/a a Python.")
4
5 if __name__ == "__main__":
6     saludo_personalizado()
```

Listing 1: Ejemplo de solución para Problema 1

```

1 def es_primo(n):
2     if n <= 1:
3         return False
4     if n <= 3:
5         return True
6     if n % 2 == 0 or n % 3 == 0:
7         return False
8     i = 5
9     while i * i <= n:
10        if n % i == 0 or n % (i + 2) == 0:
11            return False
12        i += 6
13    return True
14
15 if __name__ == "__main__":
16     numero = int(input("Ingrese un numero: "))
17     if es_primo(numero):
18         print(f"{numero} es un numero primo.")
19     else:
20         print(f"{numero} no es un numero primo.")

```

Listing 2: Ejemplo de solución para Problema 9

Rúbrica de Evaluación

Criterio	Puntos
Correcta implementación	40
Manejo de errores	20
Comentarios y documentación	15
Eficiencia del código	15
Creatividad y estilo	10
Total	100

¡Buena suerte!