

25 DE AGOSTO DE 2025

PROGRAMACIÓN

TRABAJO PRÁCTICO 1

RESUMEN

Este conjunto de problemas tiene la finalidad de introducir al entorno de programación utilizando Python. En este conjunto de problemas usted escribirá un programa de Python simple y lo entregará. Asegúrese de leer detenidamente este conjunto de problemas y las secciones dentro del documento.

CALABRESE, GERARDO NAHUEL

UNIVERSIDAD DE SAN MARTÍN

INGENIERÍA EN SISTEMAS ESPACIALES

INDICE

1.	COLABORACIÓN	1
2.	INTRODUCTORIOS	2
2.1.	INPUT Y PRINT.....	2
2.2.	STRINGS	2
2.3.	RAIZ CÚBICA	2
2.4.	COMPARACIÓN	2
2.5.	SUMAR MONTOS.....	3
3.	PROBLEMAS TÉCNICOS (NUMÉRICOS)	4
3.1.	FLUJO POR CANAL TRAPESOIDAL.....	4
3.2.	DEFLEXIÓN MÁXIMA.....	4
4.	CONTROL DE FLUJO.....	6
4.1.	BUSQUEDA DE CASA	6
4.2.	AHORRO, CON AUMENTO DE SALARIO.....	8
4.3.	LA CANTIDAD ADECUADA DE AHORRO	9

1. COLABORACIÓN

Esta permitido trabajar colaborativamente entre alumnos. Sin embargo cada estudiante debe redactar y entregar su trabajo por separado.

2. INTRODUCTORIOS

Para la resolución de los problemas se recomienda utilizar Spyder de la Distribución Anaconda. También se admite realizar la resolución en Colab u otro entorno similar, compartiendo luego un entorno en GitHub.

2.1. INPUT Y PRINT

Escribe un programa que haga lo siguiente en orden:

1. Pide al usuario que introduzca un número «x».
2. Pide al usuario que introduzca un número «y»
3. Imprime el número «x» elevado a la potencia «y».
4. Imprime el logaritmo (base 2) de «x».

2.2. STRINGS

Suponga que **s** es una cadena de caracteres en minúsculas.

Escriba un programa que cuente el número de vocales que contiene la cadena **s**. Las vocales válidas son: 'a', 'e', 'i', 'o', y 'u'. Por ejemplo, si `s = 'azcbobobegghakl'`, tu programa debe imprimir:

Número de vocales: 5

Nota: considerar que el string ingresado puede conterner letras mayúsculas.

2.3. RAIZ CÚBICA

Realice un algoritmo capaz de calcular la raíz cúbica de un número negativo o positivo mediante el método de bisección. Al encontrar la raíz, debe imprimirse en la consola una leyenda con su valor y la cantidad de pasos ejecutados hasta llegar a la solución.

2.4. COMPARACIÓN

Realice una comparación entre los métodos Bisección y Newton – Raphson para el cálculo de raíces cuadradas.

2.5. SUMAR MONTOS

Escriba un script tal que al ejecutarlo, solicite al usuario por la terminal que ingrese un número monto. Estos números deberán sumarse hasta que el usuario ingrese enter sin pasar nada (vacío). Luego, el script deberá mostrar el total en la consola.

3. PROBLEMAS TÉCNICOS (NUMÉRICOS)

3.1. FLUJO POR CANAL TRAPESOIDAL

Por un canal trapezoidal fluye agua con flujo de $Q = 20 \text{ m}^3/\text{s}$. La profundidad crítica “y” para dicho canal satisface la ecuación:

$$0 = 1 - \frac{Q^2}{g * A_c^3} * B$$

Donde $g = 9.81 \frac{\text{m}}{\text{s}^2}$, A_c es el área de la sección transversal (m^2) y B es el ancho del canal en la superficie (m). Para este caso, el ancho y el área de la sección transversal se relacionan con la profundidad “y” por medio de:

$$B = 3 + y ; A_c = 3y + \frac{y^2}{2}$$

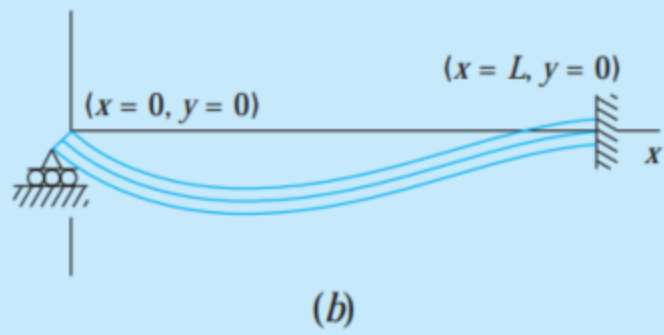
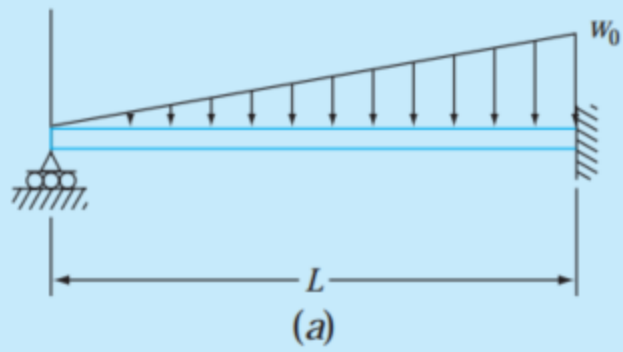
Resuelva para la profundidad crítica con el uso del método de bisección o Newton – Raphson. Haga elecciones iniciales de $x_i = 0.5$ y $x_u = 2.5$ y ejecute iteraciones hasta hasta que el error aproximado caiga por debajo de 1% o el número de iteraciones supere a 10. Analice los resultados.

3.2. DEFLEXIÓN MÁXIMA

En la figura debajo se muestra una viga uniforme sujeta a una carga distribuida linealmente creciente. La ecuación para la curva elástica resultante es:

$$y = \frac{w_0}{120EIL} * (-x^5 + 2L^2x^3 - L^4x)$$

Use bisección para determinar el punto de deflexión máxima (es decir, el valor de x cuando $dy/dx=0$). Luego substituya este valor en la ecuación dada arriba para determinar el valor de la deflexión máxima. Use los siguientes parámetros: $L = 600 \text{ mm}$, $E = 50000 \text{ kN/cm}^2$, $I=30000\text{cm}^4$ y $w_0=2.5\text{kN/cm}$.



4. CONTROL DE FLUJO

Este conjunto de problemas es una introducción al uso de control de flujo en Python y la formulación de una solución computacional a un problema. También te dará la oportunidad de explorar la búsqueda por bisección. Este conjunto de problemas tiene tres problemas.

4.1. BUSQUEDA DE CASA

Lograste titularte en la UNSAM y tenes un buen trabajo en una empresa de alta tecnología, con un ingreso estable y dolarizado. Tenes intenciones de mudarte al mejor barrio de Córdoba y decidiste que vas a empezar a ahorrar para comprar una casa. Como los precios de la vivienda son muy altos en la zona, te das cuenta de que vas a tener que ahorrar durante varios años.

En este problema, se requiere determinar cuánto tiempo tardará en ahorrar el dinero suficiente para hacer el pago inicial, teniendo en cuenta los siguientes supuestos:

1. La variable costo de la casa elegida tiene por nombre asignado **total_cost**
2. Asignar a la parte del costo necesario como pago inicial **parte_pago_inicial**. Para simplificar, supongamos que $\text{parte_pago_inicial} = 0,25$ (25%).
3. Llamemos **ahorro_actual** a la cantidad que hemos ahorrado hasta ahora. Empezas con un **ahorro_actual** de 0\$.
4. Supongamos que inviertes sabiamente tus ahorros actuales, con un rendimiento anual **r** (en otras palabras, al final de cada mes, recibes un adicional $\text{ahorro_actual} * r / 12$ para invertir en tus ahorros; el 12 se debe a que **r** es una tasa anual). Asume que tus inversiones tienen un rendimiento de **r = 0,04** (4%).
5. Asume que tu salario anual es **salario_anual**.
6. Supongamos que vas a dedicar una cierta cantidad de tu salario cada mes a ahorrar para el pago inicial. Llama a este ahora **parte_ahorrada**. Esta variable debe estar en forma decimal (es decir, 0,1 para el 10%).
7. Al final de cada mes, sus ahorros se incrementarán con el rendimiento de su inversión, más un porcentaje de tu **salario mensual** (salario anual / 12)

Escribe un programa para calcular cuántos meses tardarás en ahorrar el dinero suficiente para el pago inicial. Querrás que tus variables principales sean flotantes, por lo que deberás convertir las entradas del usuario a flotantes.

Su programa debe pedir al usuario que introduzca las siguientes variables:

1. El salario anual inicial (**salario_anual**)
2. La parte del salario que debe ahorrarse (**parte_ahorrada**)
3. El coste de la casa de sus sueños (**coste_total**)

Algunos consejos modo ayuda: etapas que probablemente deberías seguir al escribir tu código:

- Recuperar la entrada del usuario. Mira `input()` si necesitas ayuda para obtener la entrada del usuario. Para este conjunto de problemas, puedes asumir que los usuarios introducirán una entrada válida (por ejemplo, no introducirán un string cuando esperas un int)
- Inicializar algunas variables de estado. Debe decidir qué información necesita. Tenga cuidado con los valores que representan cantidades anuales y los que representan cantidades mensuales.

Pruebe diferentes entradas y vea cuánto tiempo se tarda en ahorrar para un pago inicial. Haga que su programa imprima los resultados en el formato que se muestra en los casos de prueba a continuación.

Caso de prueba 1

>>>

Introduzca su salario anual: 120000

Introduzca el porcentaje de su salario para ahorrar, como decimal: .10

Introduzca el coste de la casa de sus sueños: 1000000

Número de meses: 183

>>>

Caso de prueba 2

>>>

Introduzca su salario anual: 80000

Introduzca el porcentaje de su salario para ahorrar, como decimal: .15

Introduzca el coste de la casa de sus sueños: 500000

Número de meses: 105

>>>

4.2. AHORRO, CON AUMENTO DE SALARIO

En la sección anterior, suponíamos (de forma poco realista) que tu salario no cambiaba. Pero usted es titulado por la UNSAM, y está claro que con el tiempo va a valer más para su empresa o buscará la forma de aumentar su salario.

Copie su solución del Problema 3.1 (ya que vamos a reutilizar gran parte de ese proceso). Modifique su programa para incluir lo siguiente:

1. Haga que el usuario introduzca un aumento salarial semestral **aumento_semi_anual** (como un porcentaje decimal)
2. Después del 6º mes, aumente su salario en ese porcentaje. Repita el proceso cada seis meses.

Escriba un programa para calcular cuántos meses le llevará ahorrar el dinero suficiente para el pago inicial. Como antes, suponga que sus inversiones obtienen una rentabilidad de $r = 0,04$ (o 4%) y que el porcentaje de pago inicial requerido es de 0,25 (o 25%). Haga que el usuario introduzca las siguientes variables:

1. El salario anual inicial (**salario_anual**)
2. El porcentaje del salario que se va a ahorrar (**parte_ahorrada**)
3. El costo de la casa de tus sueños (**total_cost**)
4. El aumento semestral del salario (**aumento_semi_anual**)

Haga que suprograma imprima los resultados en el formato que se muestra en los casos de prueba a continuación.

Caso de prueba 1

>>>

Introduzca su salario anual: 120000

Introduzca el porcentaje de su salario para ahorrar, como decimal: .05

Introduzca el coste de la casa de sus sueños: 500000

Introduzca el aumento semestral, en decimales: 0.03

Número de meses: 142

>>>

Caso de prueba 2

>>>

Introduzca su salario anual: 80000

Introduzca el porcentaje de su salario para ahorrar, como decimal: .1

Introduzca el coste de la casa de sus sueños: 800000

Introduzca el aumento semestral, en decimales: 0.03

Número de meses: 159

>>>

Caso de prueba 3

>>>

Introduzca su salario anual: 75000

Introduzca el porcentaje de su salario para ahorrar, como decimal: .05

Introduzca el coste de la casa de sus sueños: 1500000

Introduzca el aumento semestral, en decimales: 0.05

Número de meses: 261

>>>

4.3. LA CANTIDAD ADECUADA DE AHORRO

En la sección anterior has tenido la oportunidad de estudiar cómo el porcentaje de tu sueldo que ahorras cada mes y tu aumento anual influyen en el tiempo que tardas en ahorrar para el pago inicial. Esto está muy bien, pero supongamos que quieres fijarte un objetivo concreto, por ejemplo, poder pagar la entrada en tres años: ¿cuánto deberías ahorrar cada mes para conseguirlo? En este problema, vas a escribir un programa para responder a esa pregunta. Para simplificar, supongamos que:

1. Tu aumento semestral es de 0,07 (7%)
2. Tus inversiones tienen una rentabilidad anual de 0,04 (4%)
3. El pago inicial es de 0,25 (25%) del coste de la casa
4. El coste de la casa para la que estás ahorrando es un penhouse en Puerto Madero con un valor de 1 millón de dólares.

Ahora vas a intentar encontrar la mejor tasa de ahorro para conseguir el pago inicial de una casa de 1 millón de dólares en 36 meses. Dado que alcanzar este objetivo es un reto, simplemente queremos que tus ahorros estén dentro de los 100\$ del pago inicial requerido. En este script, escribe un programa para calcular la mejor tasa de ahorro, en función de tu salario inicial. Deberías utilizar búsqueda por bisecciones para operar eficientemente. Llevar un registro del número de pasos que tarda la búsqueda por bisecciones es una buena práctica.

Reutiliza parte del código que escribiste para el problema anterior en este. Como estamos buscando un valor que en principio es un flotante, vamos a limitarnos a dos decimales de precisión (es decir, podemos querer ahorrar en 7.04% o 0.0704 en

decimal - pero no vamos a preocuparnos por la diferencia entre 7.041% y 7.039%). Esto significa que podemos buscar un **número entero** entre 0 y 10000 (utilizando la división entera), luego convertirlo en un porcentaje decimal para utilizarlo cuando estemos calculando el **ahorro_actual** después de 36 meses. Al utilizar este intervalo, sólo hay un número finito de números sobre los que buscamos, a diferencia del número infinito de decimales entre 0 y 1. Este intervalo ayudará a evitar bucles infinitos. La razón por la que utilizamos de 0 a 10000 es para tener en cuenta dos decimales adicionales en el rango de 0% a 100%. Los códigos deben imprimir un decimal (por ejemplo, 0,0704 para un 7,04%).

Pruebe a introducir distintos valores para su salario inicial y compruebe cómo cambia el porcentaje que debe ahorrar para alcanzar el pago inicial deseado. Tenga en cuenta también que para algunos salarios puede no ser posible ahorrar una entrada en un año y medio. En este caso, su función debe notificar al usuario que no es posible ahorrar para el pago inicial en 36 meses con una declaración impresa.

Nota: Hay varias formas correctas de implementar la búsqueda de bisección/número de pasos, por lo que sus resultados pueden no coincidir perfectamente con los del caso de prueba.

Comentarios:

- Puede haber múltiples tasas de ahorro que produzcan una cantidad de ahorro que esté dentro de los \$100 del pago inicial requerido para una casa de \$1 millón. En este caso, puede simplemente devolver cualquiera de los valores posibles.
- Dependiendo de la condición de detención y de cómo calcule un valor de prueba para la búsqueda bisección, su número de pasos puede variar ligeramente de los casos de prueba
- Tenga cuidado con la división entera cuando calcule si un porcentaje ahorrado es apropiado y cuando calcule la tasa de ahorro porcentual decimal final.
- Recuerde restablecer la(s) variable(s) apropiada(s) a sus valores iniciales para cada iteración de la búsqueda bisección.

Caso de prueba 1

>>>

Introduzca su salario anual: 150000
Tasa de ahorro máxima: 0,4411
Pasos en la búsqueda de bisección: 12

>>>

Caso de prueba 2

>>>

Introduzca su salario anual: 300000
Tasa de ahorro máxima: 0,2206
Pasos en la búsqueda de bisección: 9

>>>

Caso de prueba 3

>>>

Introduzca su salario anual: 10000
No es posible el pago total en 3 años

>>>