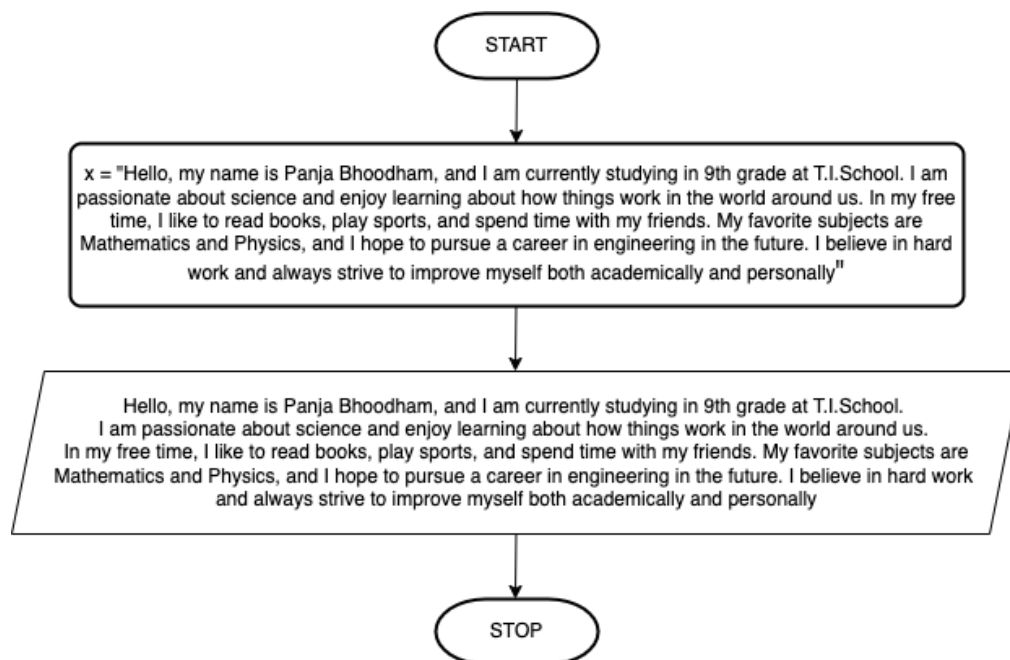


Print 5 lines about yourself using print function.

Objective: Printing 5 lines

Task: Write a python code to print 5 lines about myself using 'print' function

Python program to print the given text in double quotes



```
text = """Hello, my name is Panja Bhoodham, and I am currently studying in 9th grade at T.I.School. I am passionate about science and enjoy learning about how things work in the world around us. In my free time, I like to read books, play sports, and spend time with my friends. My favorite subjects are Mathematics and Physics, and I hope to pursue a career in engineering in the future. I believe in hard work and always strive to improve myself both academically and personally."""
```

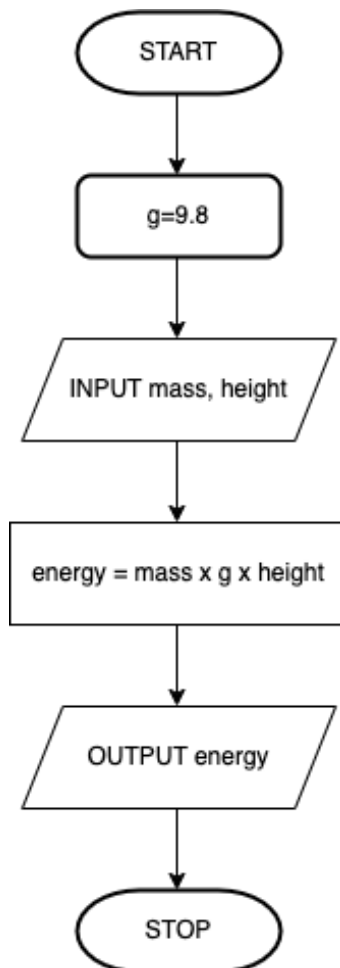
```
# Print the text
print(text)
```

Calculate energy using this formula: $\text{energy} = mgh$

Objective: Use of arithmetic operators

Task: Write a python code to calculate energy using this formula: $\text{energy} = mgh$

Program to calculate energy using the formula: $\text{energy} = mgh$



Given constant for acceleration due to gravity (in m/s^2)
 $g = 9.8$

Acceleration due to gravity in meters per second squared

Input values from the user

```
mass = float(input("Enter the mass of the object in kilograms (kg): "))
```

```

height = float(input("Enter the height in meters (m): "))

# Calculate energy using the formula: energy = m * g * h
energy = mass * g * height

# Display the result
print(f"The gravitational potential energy is: {energy} joules")

```

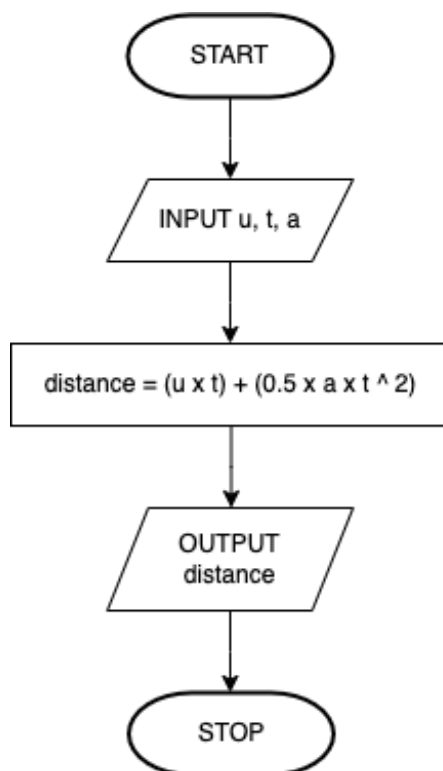
Calculate distance using the formula

$$(\text{distance} = ut + \frac{1}{2}at^2)$$

- u - is the initial velocity (in meters per second, m/s)
- t - is the time (in seconds, s)
- a - is the acceleration (in meters per second square, m/s square)
- $\frac{1}{2}at^2$ - is the term for the distance traveled due to acceleration

Objective: Use of arithmetic operators

Task: Write a program to calculate distance using this formula: **distance=ut+½ at²**



Identifiers:

- Velocity (u)
- Time (t)
- Acceleration (a)

Python program to calculate distance using the formula:

Input values from the user

```
u = float(input("Enter the initial velocity (u) in meters per second (m/s): "))
```

```
t = float(input("Enter the time (t) in seconds (s): "))
```

```
a = float(input("Enter the acceleration (a) in meters per second squared (m/s²): "))
```

Calculate distance using the formula: $\text{distance} = ut + \frac{1}{2} a t^2$

```
distance = (u * t) + (0.5 * a * t**2)
```

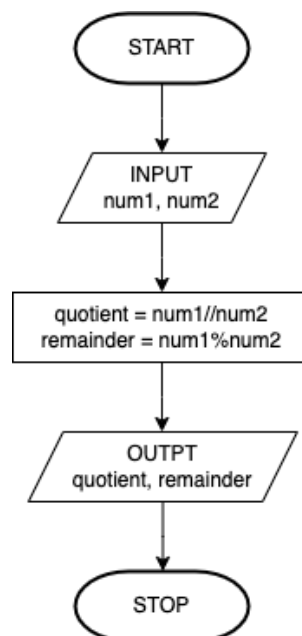
Display the result

```
print(f"The distance traveled is: {distance} meters")
```

Demonstrate the use of floor division (//) and modulo operator (%)

Objective: Use of arithmetic operators

Task: Write a program to demonstrate the use of floor division (//) and modulo operator (%) in python



Python program to demonstrate floor division (//) and modulo operator (%)

Input two numbers from the user

```
num1 = int(input("Enter the first number: "))
```

```
num2 = int(input("Enter the second number: "))

# Using floor division (//) to get the quotient
quotient = num1 // num2

# Using modulo operator (%) to get the remainder
remainder = num1 % num2

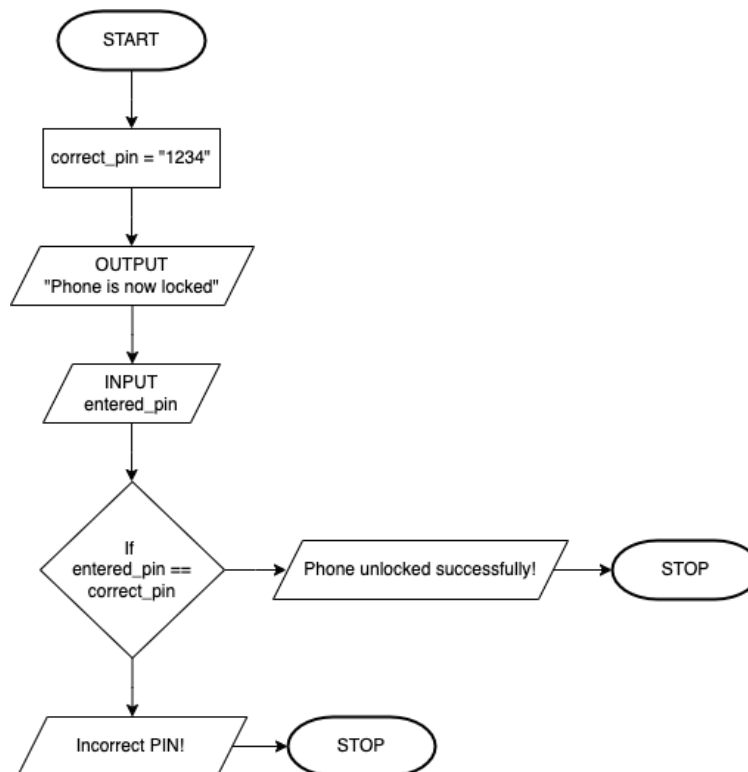
# Display the results
print(f"Floor Division of {num1} // {num2} = {quotient}")
print(f"Modulo of {num1} % {num2} = {remainder}")
```

Lock or unlock phone using pin and generate appropriate message

Objective: Use of If-else

Task: Write a program to lock or unlock phone using pin and generate appropriate message

Python program to lock and unlock a phone using PIN without functions



```
# Set the correct PIN for the phone
correct_pin = "1234"

# Lock the phone initially
print("Phone is now locked.")

# Ask the user to enter the PIN to unlock the phone
entered_pin = input("Enter your PIN to unlock the phone: ")

# Check if the entered PIN matches the correct PIN
if entered_pin == correct_pin:
    print("Phone unlocked successfully!")
else:
    print("Incorrect PIN! Phone remains locked.")
```

Title: Odd or Even Number Checker Program

Objective: The objective of this program is to determine whether a given integer is odd or even. It takes an integer input from the user and checks if the number is divisible by 2 (even) or not divisible by 2 (odd), then outputs the result.

Task:

- To determine if the number is odd or even

Task: To determine if the number is odd or even

Input: Ask user to enter a number

try:

```
number = int(input("Enter a number: "))
```

Check if the number is even or odd

```
if number % 2 == 0:
```

```
    print(f"The number {number} is Even.")
```

```
else:
```

```
    print(f"The number {number} is Odd.")
```

```
except ValueError:
```

```
    print("Invalid input! Please enter a valid integer.")
```

Title: Student Grade Calculator based on Average marks

Objective: The objective of this program is to calculate the average marks of a student based on their scores in Physics, Chemistry and Mathematics and then assign a grade according to the following criteria:

- Grade A: Average score above 90
- Grade B: Average score above 75 but less than or equal to 90
- Grade C: Average score above 65 but less than or equal to 75
- Grade D: (if the Average score is less than 65)

Task: Compute total marks and average as per given criteria, print grade

Code:

```
name=input('Enter Name:')
maths=int(input('Enter Maths Score:'))
physics=int(input('Enter Physics Score:'))
chemistry=int(input('Enter Chemistry Score:'))
avg=(maths+physics+chemistry)/3
if avg>90:
    print('Grade A')
elif (avg>75):
    print('Grade B')
elif (avg>65):
    print('Grade C')
else:
    print('Grade D')
```

Title: Kilometer to Miles Conversion Program

Objective: To write a Python program that converts a distance from kilometers to miles.

Task: Define a function for conversion, take user input, apply the conversion formula, and display the result.

Code:

```
# Python Program to Convert Kilometers to Miles

# Input: Get the distance in kilometers from the user
kilometers = float(input("Enter distance in kilometers: "))

# Conversion: 1 kilometer = 0.621371 miles
miles = kilometers * 0.621371

# Output: Display the result
```

```
print(f"{kilometers} kilometers is equal to {miles} miles.")
```

Title: Find GCD of given numbers

Objective: To find the GCD (Greatest Common Divisor) of two numbers using the Euclidean algorithm without defining a function.

Task: Accept two numbers as input, apply the Euclidean algorithm to compute the GCD, and display the result.

Code:

```
# Python Program to Find GCD of Two Numbers Without Function
```

```
# Input: Get two numbers from the user
```

```
a = int(input("Enter first number: "))
```

```
b = int(input("Enter second number: "))
```

```
# Find GCD using the Euclidean algorithm
```

```
while b != 0:
```

```
    a, b = b, a % b
```

```
# Output: Display the GCD
```

```
print(f"The GCD of the given numbers is: {a}")
```

Title: Display Calendar

Objective: To display the calendar for a specific month and year without using a custom function.

Task: Get user input for the year and month, use the calendar module to display the calendar, and print the result.

Code:

```
# Python Program to Display Calendar
```

```
import calendar
```

```
# Input: Get the year and month from the user
```

```
year = int(input("Enter year: "))
```

```
month = int(input("Enter month (1-12): "))
```



```
# Display the calendar for the given month and year
print(calendar.month(year, month))
```

Title: Adding two Matrix

Objective: To add two matrices of the same size without using a custom function.

Task: Accept matrix dimensions and elements from the user, perform element-wise addition, and display the resulting matrix.

Code:

```
# Program to add two matrices using nested loop
```

```
X = [[12,7,3],
      [4,5,6],
      [7,8,9]]
```

```
Y = [[5,8,1],
      [6,7,3],
      [4,5,9]]
```

```
result = [[0,0,0],
           [0,0,0],
           [0,0,0]]
```

```
# iterate through rows
for i in range(len(X)):
    # iterate through columns
    for j in range(len(X[0])):
        result[i][j] = X[i][j] + Y[i][j]
```

```
for r in result:
    print(r)
```

Title: Check Prime Number

Objective: To determine whether a given number is prime or not by checking divisibility without using a function.

Task: Accept a number as input, check for divisibility from 2 up to the square root of the number, and display whether it is prime or not.

Code:

Python Program to Check Whether a Given Number is Prime or Not

Input: Get a number from the user

```
num = int(input("Enter a number: "))
```

Check if the number is less than 2 (prime numbers are greater than 1)

```
if num < 2:
```

```
    print(f"{num} is not a prime number.")
```

```
else:
```

Check divisibility from 2 to the square root of the number

```
is_prime = True
```

```
for i in range(2, int(num**0.5) + 1):
```

```
    if num % i == 0:
```

```
        is_prime = False
```

```
        break
```

Output: Display the result

```
if is_prime:
```

```
    print(f"{num} is a prime number.")
```

```
else:
```

```
    print(f"{num} is not a prime number.")
```

Title: Decreasing Star Pattern

Objective: Print a pattern of stars decreasing by one each row.

Task: Use a loop to print stars in decreasing order.

Code:

```
rows = 5
```

Loop through each row

```
for i in range(rows, 0, -1):
```

Print '*' characters for each column in the current row

```
    print('*' * i)
```

Title: Science Quiz Participants

Objective: Perform list operations such as printing, deletion, and addition.

Task: Modify the list by deleting "Vikram" and adding "Jay" at the end.

Code:

```
# List of children selected for science quiz
children = ["Arjun", "Sonakshi", "Vikram", "Sandhya", "Sonal", "Isha", "Kartik"]

# Task 1: Print the whole list
print("Initial List:", children)

# Task 2: Delete the name "Vikram" from the list
children.remove("Vikram")
print("List after removing 'Vikram':", children)

# Task 3: Add the name "Jay" at the end of the list
children.append("Jay")
print("List after adding 'Jay' at the end:", children)
```

Title: Simple Interest Calculation

Objective: Calculate the simple interest based on the principal amount, rate of interest, and time period.

Task: Implement a Python program that prompts the user for input values (principal, rate of interest, and time) and calculates the simple interest using the formula.

Code:

```
# Input values: Principal, Rate of Interest, and Time
principal = float(input("Enter the principal amount: "))
rate_of_interest = float(input("Enter the rate of interest (in %): "))
time = float(input("Enter the time period (in years): "))

# Simple Interest Formula: SI = (P * R * T) / 100
simple_interest = (principal * rate_of_interest * time) / 100

# Output: Display the Simple Interest
print(f"The Simple Interest is: {simple_interest}")
```