

1) Which of the following statements is true about Python variables?

- a. Variable names can start with a number.
- b. Variables must be declared with their data types.
- c. Python variables do not need to be declared before use.
- d. Variable names can contain spaces.

Ans. c. Python variables do not need to be declared before use.

In Python, we don't need to declare variables or mention their data type before using them.

A variable is created as soon as we assign a value.

Example:

```
name = "Jyoti" # Here variable was created automatically  
age = 24      # no need to declare the type
```

So unlike some other languages, Python makes it simple.

2) As a follow-up of Q. 1., which of the following is a correct variable name in Python?

- a. 1variable
- b. variable_1
- c. variable-1
- d. variable 1

Ans. b. variable_1

Rules for Python variable names:

- Cannot start with a number (variable is wrong)
- Cannot have spaces (variable 1 is wrong)
- Cannot use - (dash) (variable1 is wrong)
- Can use letters, numbers, and _ (underscore)

3) Mention 2 ways by which you can apply comments in Python.

Ans. There are 2 ways to write comments in Python:

- **Single-line comment** → use #

```
# This is a single-line comment
```

- **Multi-line comment** → use triple quotes ''' or """

```
"""
```

This is

a multi-line

comment

```
"""
```

4) What is the difference between the 'input()' function and the 'print()' function in Python?

Ans. Difference between input() and print() in Python:

- **input()** → The **input()** function in Python always returns the user's input as a string, no matter what we type.

```
name = input("Enter your name: ")
```

- **print()** → The **print()** in Python is a function used to display/output text, numbers, or results on the screen.

```
print("Hello, world!")
```

5) How do you convert the string '123' to an integer in Python?

- a. str('123')
- b. int(123)
- c. int('123')
- d. float('123')

Ans. c. **int('123')**

- str('123') → keeps it as a string.
- int(123) → already a number, no need to convert.
- int('123') → converts the string '123' to integer 123.
- float('123') → converts to a decimal number 123.0, not an integer.

Example:

```
num = int('123')
```

```
print(num)      # 123
```

```
print(type(num)) # <class 'int'>
```

6) Differentiate between the following operators:

a. = and ==

b. / and %

c. / and //

d. * and **

If there is no difference, then mention why, and if there is a difference, then justify with examples.

Ans.

a. = and ==

- = → assignment operator (used to give a value to a variable).

`x = 5 # assigns 5 to x`

- == → comparison operator (used to check if two values are equal).

`5 == 5 # True`

`5 == 3 # False`

b. / and %

- / → division (gives the quotient as a decimal/float).

`7 / 2 # 3.5`

- % → modulus (gives the remainder).

`7 % 2 # 1`

c. / and //

- `/` → normal division (result can be a decimal).

`7 / 2 # 3.5`

- `//` → floor division (only the whole number part, no decimal).

`7 // 2 # 3`

d. * and **

- `*` → multiplication.

`5 * 2 # 10`

- `**` → exponent (power).

`5 ** 2 # 25 (5 to the power of 2)`

7) What is the output of the following Python code?

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

```
a=20
```

```
b=10
```

```
print(a+b)
```

Ans.

```
a = int(input("Enter a:")) # user enters something, e.g. 50
```

```
a = 20 # but then a is overwritten to 20
```

```
b = 10
```

```
print(a+b) # Output will be 30
```

8) What is the output of the following Python code?

```
a = "nana "
b = 8
c = a * b
print(c, "Batman")
```

Ans. `a = "nana "`

`b = 8`

`c = a * b`

`print(c, "Batman")`

- `a = "nana "` → string with a space at the end.
- `a * b` → repeats "nana " 8 times.
- So, `c = "nana nana nana nana nana nana nana nana "`
- Then `print(c, "Batman")` adds "Batman" at the end.

Output: `nana nana nana nana nana nana nana nana Batman`

9) How do you print the exact following lines in the exact order using only 1 print statement in Python?

`Hi! We are studying "Python". \n I hope you all are doing well.`

`We are going to have a great time!`

Ans. `print('Hi! We are studying "Python". \\n I hope you all are doing well.\\nWe are going to have a great time!')`

- `\n` → prints `\n` as text.
- `\n` → actually moves to the next line.

In Python, we can mix single and double quotes.

- If you start a string with single quotes '...', you can put double quotes " inside without escaping.
- If you start with double quotes "...", you can put single quotes ' inside.

10. Create a program that displays your name and complete mailing address. The address should be printed in the format that is normally used in the area where you live. Your program does not need to read any input from the user.

Ans.

Program to display name and mailing address

```
print("Jyotiranjan Parida")
print("Plot No-241/4022, Parakarana Sahi, Old Town")
print("Bhubaneswar, Odisha - 751001")
print("India")
```

Output:

Jyotiranjan Parida

Plot No-241/4022, Parakarana Sahi, Old Town

Bhubaneswar, Odisha - 751001

India

11) Given an expression $P(x) = x^{}3 + 2*x^{**}2 + 3*x+4$, try to find the value of $P(2)$ pythonically.**

Ans. $x = 2$

$$P = x^{**}3 + 2*(x^{**}2) + 3*x + 4$$

```
print("P(2) =", P)
```

Output: $P(2) = 26$

12) Write a program that begins by reading a radius, r , from the user. The program will continue by computing and displaying the area of a circle with radius r and the volume of a sphere with radius r .

Use the pi constant in the math module in your calculations.

Hint: The area of a circle is computed using the formula
 $\text{area} = \pi * r^{**}2$.

The volume of a sphere is computed using the formula
 $\text{volume} = 4/3 * \pi * r^{**}3$.

Ans.

- Using **math** module

```
import math
```

```
# Read radius from user
```

```
r = float(input("Enter the radius: "))
```

```
# Calculate area of circle
```

```
area = math.pi * r**2
```

```
# Calculate volume of sphere
```

```
volume = (4/3) * math.pi * r**3  
# Display results  
print("Area of the circle:", area)  
print("Volume of the sphere:", volume)
```

OUTPUT:

5.2

Area of the circle: 84.94866535306801
Volume of the sphere: 588.9774131146048

- Without Using **math** Module:

```
# Approximate value of pi  
pi = 3.1416  
# Read radius from user  
r = float(input("Enter the radius: "))  
# Calculate area of circle  
area = pi * r**2  
# Calculate volume of sphere  
volume = (4/3) * pi * r**3
```

```
# Display results  
print("Area of the circle:", area)
```

```
print("Volume of the sphere:", volume)
```

OUTPUT:

5.2

Area of the circle: 84.948864

Volume of the sphere: 588.9787904

- 13) Write a program that reads a positive integers ,n, from the user and then display the sum of all of the integers from 1 to n. The sum of the first n positive integers can be computed using the formula:**

$$\text{Sum} = \frac{(n)(n+1)}{2}$$

Ans.

```
n = int(input("Enter a positive integer: "))
```

```
# Calculate the sum using the formula
```

```
sum_of_integers = (n * (n + 1)) // 2 # use // to get integer result
```

Display the result

```
print("The sum of integers from 1 to", n, "is:", sum_of_integers)
```

OUTPUT: 10

The sum of integers from 1 to 10 is: 55

14) Evaluate the following expressions:

(a<b) or (not(c==b) and (c<a))

a. a =10, b=5, c=0

b. a=1.21, b=1.20, c=1.22

Ans:

The expression is: **(a<b) or (not(c==b) and (c<a))**

a. (a<b) or (not(c==b) and (c<a))

a < b → 10 < 5 → False

c == b → 0 == 5 → False → not(c == b) → True

c < a → 0 < 10 → True

(a<b) or (not(c==b)) → False or True → True

(not(c==b) and (c<a)) → True and True → True

(a<b) or (not(c==b) and (c<a)) :

False or True and True → True

Result: True

b. a = 1.21, b = 1.20, c = 1.22

a < b → 1.21 < 1.20 → False

c == b → 1.22 == 1.20 → False → not(c==b) → True

c < a → 1.22 < 1.21 → False

(not(c==b) and (c<a)) → True and False → False

$(a < b) \text{ or } (\text{not}(c == b) \text{ and } (c < a))$:

False or True and False → False

Result: False

15) Assume you start investing in Mutual Funds with Rs. 1000 and plan to leave your money invested.

Calculate and display how much money you will have after 10, 20 and 30 years. Use the following formula for determining these amounts:

$$a = p(1+r)n$$

where p (principal) = Rs. 1000,

r (annual rate of return) = 12

n (number of years) = 10, 20, 30,

a (amount on deposit at the end of the n th year).

Disclaimer: Investing in Mutual Funds is subject to Market Risks. Do your due diligence before investing.

Ans:

$$p = 1000$$

Annual rate of return in decimal

$$r = 12 / 100 \quad \# 12\% \rightarrow 0.12$$

Number of years

$$\text{years} = [10, 20, 30]$$

```
# Calculate and display amount for each period  
for n in years:
```

$$a = p * (1 + r)^n$$

```
print(f"Amount after {n} years: Rs. {a:.2f}")
```

`:.2f` → formats the output to 2 decimal places.

OUTPUT:

Amount after 10 years: Rs. 3105.85

Amount after 20 years: Rs. 9646.29

Amount after 30 years: Rs. 29959.92

16) Import relevant Python modules and print the values of e^π and π^e . Then, if $e^\pi > \pi^e$, print "ok". Otherwise print "ok anyway".

Ans.

```
import math
```

```
# Calculate  $e^\pi$  and  $\pi^e$ 
```

```
e_power_pi = math.e ** math.pi
```

```
pi_power_e = math.pi ** math.e
```

```
# Print the values
```

```
print("e^\pi =", e_power_pi)
```

```
print("\pi^e =", pi_power_e)
```

```
# Compare and print the message
```

```
if e_power_pi > pi_power_e:
```

```
    print("ok")
```

```
else:
```

```
    print("ok anyway")
```

OUTPUT:

$e^\pi = 23.140692632779263$

$\pi^e = 22.45915771836104$

ok

17) Evaluate the following expressions involving arithmetic operators:

a. $-7 * 20 + 8 / 16 * 2 + 54$

b. $7^{**}2 // 9 \% 3$

c. $(7 - 4 * 2) * 10 - 25 * 8 // 5$

d. $5 \% 10 + 10 - 25 * 8 // 5$

Ans: a. $-7 * 20 + 8 / 16 * 2 + 54$

Mathematical step-by-step:

- $-7 * 20 = -140$

- $8 / 16 = 0.5 \rightarrow 0.5 * 2 = 1.0$

- $-140 + 1.0 + 54 = -85.0$

In Python

```
result_a = -7 * 20 + 8 / 16 * 2 + 54
```

```
print(result_a) # Output: -85.0
```

b. $7^{**}2 // 9 \% 3$

Mathematical step-by-step:

- $7^{**}2 = 49$
- $49 // 9 = 5$ (floor division)
- $5 \% 3 = 2$

In Python

```
result_b = 7**2 // 9 % 3
```

```
print(result_b) # Output: 2
```

c. $(7 - 4 * 2) * 10 - 25 * 8 // 5$

Mathematical step-by-step:

- $4 * 2 = 8$
- $7 - 8 = -1$
- $(-1) * 10 = -10$
- $25 * 8 = 200$
- $200 // 5 = 40$
- $-10 - 40 = -50$

In Python Code:

```
result_c = (7 - 4*2) * 10 - 25*8 // 5
```

```
print(result_c) # Output: -50
```

d. $5 \% 10 + 10 - 25 * 8 // 5$

Mathematical step-by-step:

- $5 \% 10 = 5$ (remainder of $5 \div 10$)
- $25*8 = 200$
- $200 // 5 = 40$
- $5 + 10 - 40 = -25$

In Python Code:

```
result_d = 5 % 10 + 10 - 25*8 // 5
```

```
print(result_d) # Output: -25
```

18) Evaluate the following expressions involving relational and logical operators:

- a. 'hi' > 'hello' and 'bye' < 'Bye'
- b. 'hi' > 'hello' or 'bye' < 'Bye'
- c. 7 > 8 or 5 < 6 and 'Iamfine' > 'Iamnotfine'
- d. 10 != 9 and 29 >= 29
- e. 10 != 9 and 29 >= 29 and 'hi' > 'hello' or 'bye' < 'Bye' and 7 <= 2.5

f. $5 \% 10 + 10 < 50 \text{ and } 29 > = 29$

g. $7 ** 2 <= 5 // 9 \% 3 \text{ or } \text{'bye'} < \text{'Bye'}$

h. $5 \% 10 < 8 \text{ and } -25 > 1 * 8 // 5$

i. $7 ** 2 // 4 + 5 > 8 \text{ or } 5 != 6$

j. $7 / 4 < 6 \text{ and } \text{'I am fine'} > \text{'I am not fine'}$

k. $10 + 6 * 2 ** 2 != 9 // 4 - 3 \text{ and } 29 > = 29 / 9$

l. $\text{'hello'} * 5 > \text{'hello'} \text{ or } \text{'bye'} < \text{'Bye'}$

Ans:

a. $\text{'hi'} > \text{'hello'} \text{ and } \text{'bye'} < \text{'Bye'}$

Step by step:

- $\text{'hi'} > \text{'hello'} \rightarrow \text{compares lexicographically} \rightarrow \text{'h'} > \text{'h'}?$
check next $\rightarrow \text{'i'} > \text{'e'} \rightarrow \text{True}$
- $\text{'bye'} < \text{'Bye'} \rightarrow \text{lowercase 'b'} > \text{uppercase 'B' in ASCII}$
 $\rightarrow \text{False}$
- **True and False $\rightarrow \text{False}$**

Result: False

b. $\text{'hi'} > \text{'hello'} \text{ or } \text{'bye'} < \text{'Bye'}$

Step by step:

- $\text{'hi'} > \text{'hello'} \rightarrow \text{True}$
- $\text{'bye'} < \text{'Bye'} \rightarrow \text{False}$
- **True or False $\rightarrow \text{True}$**

Result: True

c. $7 > 8$ or $5 < 6$ and 'Iamfine' > 'Iamnotfine'

Step by step:

- $7 > 8 \rightarrow \text{False}$
- $5 < 6 \rightarrow \text{True}$
- 'Iamfine' > 'Iamnotfine' \rightarrow 'f' > 'n'? \rightarrow False
- $5 < 6$ and 'Iamfine' > 'Iamnotfine' \rightarrow True and False \rightarrow False
- $7 > 8$ or (...) \rightarrow False or False \rightarrow False

Result: False

d. $10 \neq 9$ and $29 \geq 29$

Step by step:

- $10 \neq 9 \rightarrow \text{True}$
- $29 \geq 29 \rightarrow \text{True}$

True and True \rightarrow True

Result: True

e. $10 \neq 9$ and $29 \geq 29$ and 'hi' > 'hello' or 'bye' < 'Bye' and $7 \leq 2.5$

Step by step (consider precedence: and before or):

- $10 \neq 9 \rightarrow \text{True}$
- $29 \geq 29 \rightarrow \text{True}$
- 'hi' > 'hello' → True
- 'bye' < 'Bye' → False
- $7 \leq 2.5 \rightarrow \text{False}$

Here and have the highest precedence than or.

- $10 \neq 9$ and $29 \geq 29$ and 'hi' > 'hello' → True and True and True → True
- 'bye' < 'Bye' and $7 \leq 2.5 \rightarrow \text{False and False} \rightarrow \text{False}$
- Full expression → True or False → True

Result: True

f. $5 \% 10 + 10 < 50$ and $29 \geq 29$

Step by step:

- $5 \% 10 + 10 \rightarrow 5 + 10 \rightarrow 15$
- $15 < 50 \rightarrow \text{True}$
- $29 \geq 29 \rightarrow \text{True}$

True and True → True

Result: True

g. $7 \text{ ** } 2 \leq 5 // 9 \% 3$ or 'bye' < 'Bye'

Step by step:

- $7 \text{ ** } 2 = 49$
- $5 // 9 = 0$
- $0 \% 3 = 0$
- $49 \leq 0 \rightarrow \text{False}$
- 'bye' < 'Bye' $\rightarrow \text{False}$

False or False $\rightarrow \text{False}$

Result: False

h. $5 \% 10 < 8$ and $-25 > 1 * 8 // 5$

Step by step:

- $5 \% 10 = 5 \rightarrow 5 < 8 \rightarrow \text{True}$
- $1 * 8 // 5 = 8 // 5 = 1$
- $-25 > 1 \rightarrow \text{False}$

True and False $\rightarrow \text{False}$

Result: False

i. $7 ** 2 // 4 + 5 > 8 \text{ or } 5 != 6$

Step by step:

- $7 ** 2 = 49$
- $49 // 4 = 12$
- $12 + 5 = 17$
- $17 > 8 \rightarrow \text{True}$
- $5 != 6 \rightarrow \text{True}$

True or True → True

Result: True

j. $7/4 < 6$ and 'I am fine' > 'I am not fine'

Step by step:

- $7 / 4 = 1.75 \rightarrow 1.75 < 6 \rightarrow \text{True}$
- 'I am fine' > 'I am not fine' → compares lexicographically → 'f' > 'n' → False

True and False → False

Result: False

k. $10 + 6 * 2^{}2 != 9 // 4 - 3$ and $29 >= 29 / 9$**

Step by step:

- $2^{**}2 = 4$
- $6 * 4 = 24$
- $10 + 24 = 34$
- $9 // 4 = 2 \rightarrow 2 - 3 = -1$
- $34 != -1 \rightarrow \text{True}$
- $29 / 9 \approx 3.222 \rightarrow 29 >= 3.222 \rightarrow \text{True}$

True and True \rightarrow True

Result: True

l. $'hello' * 5 > 'hello' \text{ or } 'bye' < 'Bye'$

Step by step:

- $'hello' * 5 = \text{'hellohellohellohellohello'}$
- $\text{'hellohellohellohellohello'} > \text{'hello'} \rightarrow \text{True}$
- $\text{'bye'} < \text{'Bye'} \rightarrow \text{False}$

True or False \rightarrow True

Result: True

19) The program that you create for this exercise will begin by reading the cost of a meal ordered at a restaurant from the user. Then your program will compute the tax and tip for the meal. Use your local tax rate when computing the amount of tax owing. Compute the tip as 18 percent of the meal amount (without the tax). The output from your program should include the tax amount, the tip amount, and the grand total for the meal including both the tax and the tip. Format the output so that all of the values are displayed using two decimal places.

Ans.

How it works:

- **meal_cost** → user enters the meal price.
- **tax = meal_cost * tax_rate** → calculates tax.
- **tip = meal_cost * 0.18** → calculates 18% tip on the meal.
- **total = meal_cost + tax + tip** → total cost including tax and tip.
- **:.2f** → formats the output to 2 decimal places.

```
meal_cost = float(input("Enter the cost of the meal: Rs. "))
```

```
# Set your local tax rate (for example, 5% = 0.05)
```

```
tax_rate = 0.05 # change this to your local tax rate
```

```
# Calculate tax amount
```

```
tax = meal_cost * tax_rate
```

Calculate tip (18% of meal cost, without tax)

```
tip = meal_cost * 0.18
```

Calculate total

```
total = meal_cost + tax + tip
```

Display the results with 2 decimal places

```
print(f"Tax amount: Rs. {tax:.2f}")
```

```
print(f"Tip amount: Rs. {tip:.2f}")
```

```
print(f"Grand total: Rs. {total:.2f}")
```

OUTPUT:

Enter the cost of the meal: Rs. 1000

Tax amount: Rs. 50.00

Tip amount: Rs. 180.00

Grand total: Rs. 1230.00

20) Create a program that reads two integers, a and b, from the user. Your program should compute and display:

- **The sum of a and b**
- **The difference when b is subtracted from a**
- **The product of a and b**
- **The quotient when a is divided by b**
- **The remainder when a is divided by b**
- **The result of $\log_{10} a$**

- **The result of ab**

Hint: You will probably find the `log10` function in the `math` module helpful for computing the second last item in the list.

Ans.

```
import math
```

```
a = int(input("Enter the first integer (a): "))
```

```
b = int(input("Enter the second integer (b): "))
```

Sum

```
sum_ab = a + b
```

Difference (a - b)

```
diff_ab = a - b
```

Product

```
prod_ab = a * b
```

Quotient (a / b)

```
if b != 0:
```

```
    quot_ab = a / b
```

```
else:
```

```
    quot_ab = "Undefined (division by zero)"
```

Remainder (a % b)

if b != 0:

rem_ab = a % b

else:

rem_ab = "Undefined (division by zero)"

Log base 10 of a

if a > 0:

log_a = math.log10(a)

else:

log_a = "Undefined (log10 of non-positive number)"

a raised to the power b

power_ab = a ** b

Display the results

print(f"Sum: {sum_ab}")

print(f"Difference (a - b): {diff_ab}")

print(f"Product: {prod_ab}")

print(f"Quotient (a / b): {quot_ab}")

print(f"Remainder (a % b): {rem_ab}")

```
print(f"log10(a): {log_a}")
```

```
print(f"a^b: {power_ab}")
```

OUTPUT:

Enter the first integer (a):=100

Enter the second integer (b):=5

Sum: 105

Difference (a - b): 95

Product: 500

Quotient (a / b): 20.0

Remainder (a % b): 0

log10(a): 2.0

a^b: 10000000000

- 21) Write a program that determines how quickly an object is travelling when it hits the ground. The user will enter the height from which the object is dropped in meters (m). Because the object is dropped its initial speed is 0 m/s. Assume that the acceleration due to gravity is 9.8m/s². You can use the formula

$$v_f = (v_i^2 + 2ad)^{1/2}$$

to compute the final speed, v_f , when the initial speed, v_i , acceleration, a , and distance, d , are known.

Ans.

$vi = 0$ (initial speed)

$a = 9.8 \text{ m/s}^2$ (acceleration due to gravity)

d= height (entered by the user)

```
import math
```

```
d = float(input("Enter the height from which the object is dropped (in meters): "))
```

Constants

$vi = 0 \quad \# \text{ initial speed}$

$a = 9.8 \quad \# \text{ acceleration due to gravity}$

Calculate final speed

```
vf = math.sqrt(vi**2 + 2 * a * d)
```

```
print(f"The object will be travelling at {vf:.2f} m/s when it hits the ground.")
```

OUTPUT:

Enter the height from which the object is dropped (in meters): 5.5

The object will be travelling at 10.38 m/s when it hits the ground.

22) Write a program that begins by reading a temperature from the user in degrees Celsius. Then your program should display the equivalent temperature in degrees Fahrenheit and degrees Kelvin. The calculations needed to convert between different units of temperature can be found on the Internet.

Ans.

Formulas are:

- **Fahrenheit ($^{\circ}\text{F}$):**

$$\text{F} = \frac{9}{5} * \text{C} + 32$$

- **Kelvin(K)**

$$\text{K} = \text{C} + 273.$$

Temperature Conversion Formulas 

Celsius to Kelvin: $\text{K} = \text{C} + 273.15$

Kelvin to Celcius: $\text{C} = \text{K} - 273.15$

Fahrenheit to Celsius: $\text{C} = (\text{F} - 32) * \frac{5}{9}$

Celsius to Fahrenheit: $\text{F} = \text{C} \left(\frac{9}{5} \right) + 32$

Fahrenheit to Kelvin: $\text{K} = (\text{F} - 32) * \frac{5}{9} + 273.15$

Kelvin to Fahrenheit: $\text{F} = (\text{K} - 273.15) * \frac{9}{5} + 32$

```
celsius = float(input("Enter temperature in Celsius: "))
```

Convert to Fahrenheit

```
fahrenheit = (celsius * 9/5) + 32
```

Convert to Kelvin

```
kelvin = celsius + 273.15
```

```
print(f"Temperature in Fahrenheit: {fahrenheit:.2f} °F")
```

```
print(f"Temperature in Kelvin: {kelvin:.2f} K")
```

OUTPUT:

Enter temperature in Celsius: 50.5

Temperature in Fahrenheit: 122.90 °F

Temperature in Kelvin: 323.65 K

23) Write a program that reads a four-digit integer from the user and displays the sum of its digits.

For example : if the user enters 3141 then your program should display $3 + 1 + 4 + 1 = 9$.

Ans.

```
num = int(input("Enter a four-digit integer: "))
```

Extract digits and find their sum

```
digit_sum = 0
```

```
for digit in str(num):
```

```
    digit_sum += int(digit)
```

Display result

```
print("The sum of digits is:", digit_sum)
```

OUTPUT:

Enter a four-digit integer: **3141**

The sum of digits is: 9

What each line does (with num = **3141**)

1. num = int(input(...))

- User types 3141 → input() returns the string "3141".
- int("3141") converts it to the number 3141.
- So now num == 3141.

2. digit_sum = 0

- We start a running total called digit_sum, initially 0.

3. for digit in str(num):

- str(num) makes "3141" (a string).
- The for loop goes through each character of "3141" one by one: '3', '1', '4', '1'.

4. digit_sum += int(digit)

- Each loop: convert the character digit back to an integer with int(digit) and add it to digit_sum.

24) Write a program that reads three integers from the user and displays them in sorted order (from smallest to largest). Use the min and max functions to find the smallest and largest values. The middle value can be found by computing the sum of all three values, and then subtracting the minimum value and the maximum value.

Ans. If user enters 15, 8, 20:

$$\text{smallest} = \min(15, 8, 20) = 8$$

$$\text{largest} = \max(15, 8, 20) = 20$$

$$\text{middle} = (15 + 8 + 20) - (8 + 20) = 43 - 28 = 15$$

OUTPUT: Numbers in sorted order: 8 15 20

```
a = int(input("Enter the first integer: "))
```

```
b = int(input("Enter the second integer: "))
```

```
c = int(input("Enter the third integer: "))
```

Find smallest and largest

```
smallest = min(a, b, c)
```

```
largest = max(a, b, c)
```

Middle value = total sum - (smallest + largest)

```
middle = (a + b + c) - (smallest + largest)
```

Display sorted order

```
print("Numbers in sorted order:", smallest, middle, largest)
```

OUTPUT:

Enter the first integer: 15

Enter the second integer: 8

Enter the third integer: 20

Numbers in sorted order: 8 15 20

25) Create a program that reads duration from the user as a number of days, hours, minutes, and seconds.

Compute and display the total number of seconds represented by this duration.

Ans.

We know:

- $1 \text{ day} = 24 \times 60 \times 60 = 86400 \text{ seconds}$
- $1 \text{ hour} = 60 \times 60 = 3600 \text{ seconds}$
- $1 \text{ minute} = 60 \text{ seconds}$

So total seconds =

days * 86400 + hours * 3600 + minutes * 60 + seconds

```
days = int(input("Enter number of days: "))
```

```
hours = int(input("Enter number of hours: "))
```

```
minutes = int(input("Enter number of minutes: "))
```

```
seconds = int(input("Enter number of seconds: "))
```

Compute total seconds

```
total_seconds = days * 86400 + hours * 3600 + minutes * 60 +  
seconds
```

Display the result

```
print(f"The total duration in seconds is: {total_seconds}")
```

OUTPUT:

Enter number of days: 1

Enter number of hours: 2

Enter number of minutes: 30

Enter number of seconds: 15

The total duration in seconds is: 95415

In Mathematics Steps:

$$1*86400 + 2*3600 + 30*60 + 15$$

$$= 86400 + 7200 + 1800 + 15$$

$$= 95415$$