Assignmnent 9

- Conditional statemnts Q&A

EASY LEVEL (1-20)

1. Check if a number is even or odd

```
num = int(input("Enter a number: "))
if num % 2 == 0:
 print("Even")
else:
 print("Odd")
```

```
Enter a number:  10000000534
```

```
Even
```

2. Check if a person is eligible to vote (age 18 or above)

```
age = int(input("Enter age: "))
if age >= 18:
 print("Eligible to vote")
else:
 print("Not eligible to vote")
```

```
Enter age:  35
```

```
Eligible to vote
```

3. Determine if a given year is a leap year or not

```
year = int(input("Enter a year: "))
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
 print("Leap Year")
else:
 print("Not a Leap Year")
```

```
Enter a year:  2000

Leap Year
```

4.Check if a number is positive, negative, or zero

```
num = int(input("Enter a number: "))
if num > 0:
 print("Positive")
elif num < 0:
 print("Negative")
else:
 print("Zero")
```

```
Enter a number:  5

Positive
```

5. Write a program to find the greatest of two numbers

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
if a > b:
 print("Greatest:", a)
else:
 print("Greatest:", b)
```

```
Enter first number:  25
Enter second number:  28

Greatest: 28
```

6. Determine if a number is a multiple of 5

```
num = int(input("Enter a number: "))
if num % 5 == 0:
 print("Multiple of 5")
else:
 print("Not a multiple of 5")
```

```
Enter a number:  125
```

```
Multiple of 5
```

7. Check if a character is a vowel or consonant

```
char = input("Enter a character: ").lower()
if char in 'aeiou':
 print("Vowel")
else:
 print("Consonant")
```

```
Enter a character:  i
```

```
Vowel
```

8. Determine if a person is eligible for a senior citizen discount (age 60+)

```
age = int(input("Enter age: "))
if age >= 60:
 print("Eligible for senior citizen discount")
else:
 print("Not eligible")
```

```
Enter age:  85
```

```
Eligible for senior citizen discount
```

9. Check if a number is a single-digit number

```
num = int(input("Enter a number: "))
if 0 <= abs(num) < 10:      # The abs(num) function in Python returns the absolute value of a number.
 print("Single-digit number")
```

```
else:
 print("Not a single-digit number")
```

```
Enter a number:  33

Not a single-digit number
```

10. Print "Good Morning" if the time is before 12 PM, otherwise print "Good Afternoon"

```
hour = int(input("Enter hour (24-hour format): "))
if hour < 12:
 print("Good Morning")
else:
 print("Good Afternoon")
```

```
Enter hour (24-hour format):  24

Good Afternoon
```

11. Check if a string is empty or not

```
s = input("Enter a string: ")
if not s:
 print("String is empty")
else:
 print("String is not empty")
```

```
Enter a string:

String is empty
```

12. Verify if a number is a perfect square

```
import math
num = int(input("Enter a number: "))
if math.isqrt(num) ** 2 == num:
 print("Perfect Square")
else:
 print("Not a perfect square")
```

```
Enter a number:  225

Perfect Square
```

13. Determine if a number is between 1 and 100

```
num = int(input("Enter a number: "))
if 1 <= num <= 100:
 print("Within range")
else:
 print("Out of range")
```

```
Enter a number:  82

Within range
```

14. Print "Weekend" if the day is Saturday or Sunday; otherwise, print "Weekday"

```
day = input("Enter a day: ").lower() #The .lower() method in Python
#                                     is used to convert all
characters in a string to lowercase.
if day in ["saturday", "sunday"]:
 print("Weekend")
else:
 print("Weekday")
```

```
Enter a day:  sunday

Weekend
```

15. Find if a given number is exactly divisible by both 3 and 7

```
num = int(input("Enter a number: "))
if num % 3 == 0 and num % 7 == 0:
 print("Divisible by 3 and 7")
else:
 print("Not divisible by both")
```

```
Enter a number:  149

Not divisible by both
```

16. Check if the sum of two numbers is greater than 100

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
if (a + b) > 100:
 print("Sum is greater than 100")
else:
 print("Sum is 100 or less")
```

```
Enter first number:  25
Enter second number:  87

Sum is greater than 100
```

17. Write a program to find the minimum of two numbers

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
if a < b:
 print("Minimum:", a)
else:
 print("Minimum:", b)
```

```
Enter first number:  25
Enter second number:  56

Minimum: 25
```

18. Check if a number is divisible by 2 but not by 3

```
num = int(input("Enter a number: "))
if num % 2 == 0 and num % 3 != 0:
 print("Divisible by 2 but not by 3")
else:
 print("Does not meet criteria")
```

```
Enter a number:  23

Does not meet criteria
```

19. Determine if a given alphabet is uppercase or lowercase

```
char = input("Enter an alphabet: ")
if char.isupper():
 print("Uppercase")
else:
 print("Lowercase")
```

```
Enter an alphabet:  MAHESH

Uppercase
```

20. Check if a triangle is valid given three side lengths

```
a = int(input("Enter first side: "))
b = int(input("Enter second side: "))
c = int(input("Enter third side: "))
if a + b > c and a + c > b and b + c > a:
 print("Valid Triangle")
else:
    print("Invalid Triangle")
```

```
Enter first side:  3
Enter second side:  4
Enter third side:  5

Valid Triangle
```

MEDIUM LEVEL (20-40)

21. Find the largest of three numbers

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
c = int(input("Enter third number: "))
if a >= b and a >= c:
 print("Largest:", a)
elif b >= a and b >= c:
 print("Largest:", b)
else:
 print("Largest:", c)
```

```
Enter first number:  15
Enter second number:  35
Enter third number:  95

Largest: 95
```

22. Determine if a number is a prime number

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

if num > 1:
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            print("Not a prime number")
            break
    else:
        print("Prime number")
else:
    print("Not a prime number")
```

```
Enter a number:  13

Prime number
```

23. Check if a person is eligible for a driving license

```
age = int(input("Enter age: "))
passed_test = input("Did you pass the driving test? (yes/no):
").lower()
if age >= 18 and passed_test == "yes":
 print("Eligible for a driving license")
else:
 print("Not eligible")
```

```
Enter age:  15
Did you pass the driving test? (yes/no):  yes

Not eligible
```

24. Determine if a triangle is equilateral, isosceles, or scalene

```
a = int(input("Enter first side: "))
b = int(input("Enter second side: "))
c = int(input("Enter third side: "))
if a == b == c:
 print("Equilateral Triangle")
elif a == b or b == c or a == c:
 print("Isosceles Triangle")
else:
 print("Scalene Triangle")
```

```
Enter first side:  3
Enter second side:  4
Enter third side:  5

Scalene Triangle
```

25. Determine if a student passes or fails

```
marks = int(input("Enter marks: "))
if marks >= 40:
 print("Pass")
else:
 print("Fail")
```

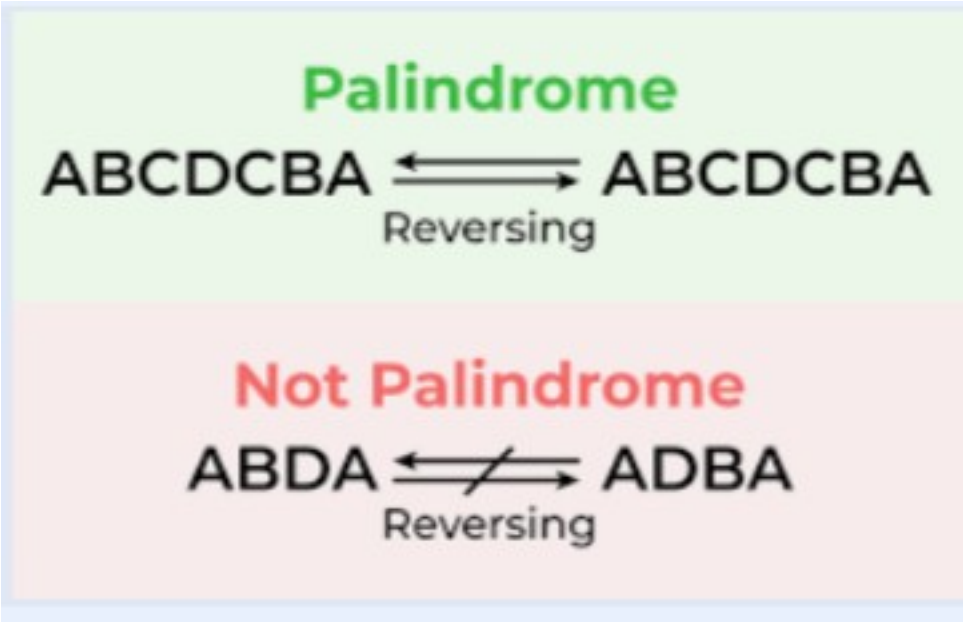
```
Enter marks:  35
```

```
Fail
```

26. Check if a number is a palindrome

```
num = input("Enter a number: ")
if num == num[::-1]:
 print("Palindrome")
else:
 print("Not a Palindrome")
```

```
Enter a number:  anna

Palindrome
```



27. Calculate electricity bill

```
units = int(input("Enter electricity units consumed: "))
if units <= 100:
 bill = units * 5
elif units <= 300:
 bill = (100 * 5) + (units - 100) * 10
else:
 bill = (100 * 5) + (200 * 10) + (units - 300) * 15
print("Total Bill: ₹", bill)
```

```
Enter electricity units consumed:  163

Total Bill: ₹ 1130
```

28. Find the grade of a student

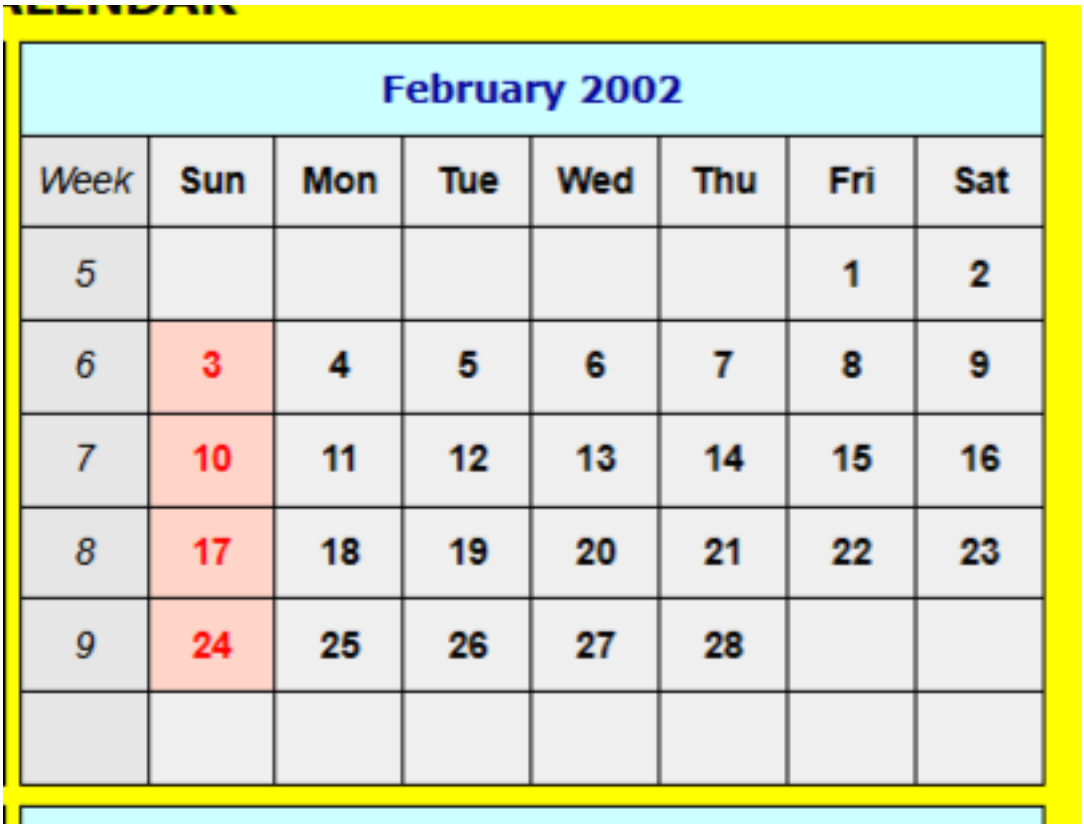
```
marks = int(input("Enter marks: "))
if marks >= 90:
 print("Grade: A")
elif marks >= 80:
 print("Grade: B")
elif marks >= 70:
 print("Grade: C")
elif marks >= 60:
 print("Grade: D")
elif marks >= 40:
 print("Grade: E")
else:
 print("Grade: F (Fail)")
```

```
Enter marks:  70

Grade: C
```

29. Determine if a given date is valid

```
import calendar
day = int(input("Enter day: "))
month = int(input("Enter month: "))
year = int(input("Enter year: "))
if 1 <= month <= 12 and 1 <= day <= calendar.monthrange(year, month)
[1]:
    #is used to get the number of days in a given month and year,
using the calendar module.
 print("Valid date")
else:
 print("Invalid date")
print("")
```

```
Enter day:  29
Enter month:  2
Enter year:  2002

Invalid date
```



February 2002

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
5						1	2
6	3	4	5	6	7	8	9
7	10	11	12	13	14	15	16
8	17	18	19	20	21	22	23
9	24	25	26	27	28		

30. Check if a given time is AM or PM

```
hour = int(input("Enter hour (24-hour format): "))
if hour < 12:
 print("AM")
else:
 print("PM")
```

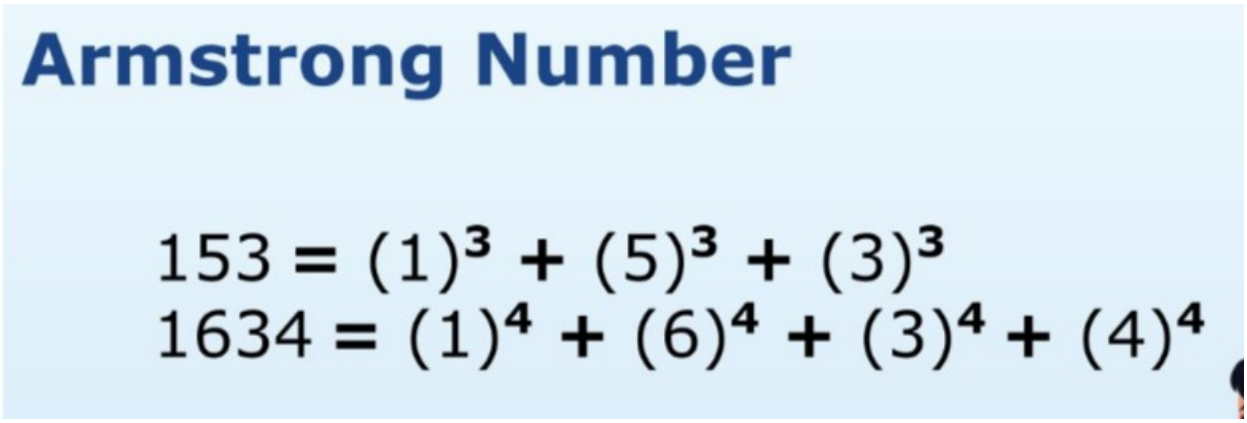
```
Enter hour (24-hour format):  13

PM
```

31. Check if a number is an Armstrong number □ □□□□

```
num = input("Enter a number: ")
power = len(num)

if sum(int(digit) ** power for digit in num) == int(num):
 print("Armstrong Number")
```

```
else:
 print("Not an Armstrong Number")
```

```
Enter a number:  54748

Armstrong Number
```



- **3-digit Armstrong numbers:**

python

```
153 → 1³ + 5³ + 3³ = 1 + 125 + 27 = 153 ✅
370 → 3³ + 7³ + 0³ = 27 + 343 + 0 = 370 ✅
371 → 3³ + 7³ + 1³ = 27 + 343 + 1 = 371 ✅
407 → 4³ + 0³ + 7³ = 64 + 0 + 343 = 407 ✅
```

- **4-digit Armstrong numbers:**

python

```
1634 → 1⁴ + 6⁴ + 3⁴ + 4⁴ = 1 + 1296 + 81 + 256 = 1634 ✅
8208 → 8⁴ + 2⁴ + 0⁴ + 8⁴ = 4096 + 16 + 0 + 4096 = 8208 ✅
9474 → 9⁴ + 4⁴ + 7⁴ + 4⁴ = 6561 + 256 + 2401 + 256 = 9474 ✅
```

🧠 Formula:

For a number with `n` digits:

python Copy Edit

```python
Armstrong if: sum(int(digit)**n for digit in str(number)) == number
```

✅ Full Example:

python

```python
def is_armstrong(number):
    n = len(str(number))  # Number of digits
    total = sum(int(digit) ** n for digit in str(number))
    return total == number

# Test
num = int(input("Enter a number: "))
if is_armstrong(num):
    print(f"{num} is an Armstrong number ✅")
else:
    print(f"{num} is not an Armstrong number ❌")
```

🔍 Explanation (example: 54748):

Digits: 5, 4, 7, 4, 8

Power: 5 (because it's a 5-digit number)

python

```
5⁵ + 4⁵ + 7⁵ + 4⁵ + 8⁵ =
3125 + 1024 + 16807 + 1024 + 32768 = 54748 ✅
```

32. Determine the type of quadrilateral

```
a = int(input("Enter first side: "))
b = int(input("Enter second side: "))
c = int(input("Enter third side: "))
d = int(input("Enter fourth side: "))
if a == b == c == d:
 print("Square")
elif a == c and b == d:
 print("Rectangle")
else:
 print("Other Quadrilateral")
```

```
Enter first side:  1
Enter second side:  2
Enter third side:  3
Enter fourth side:  4

Other Quadrilateral
```

33. Implement a basic calculator □□□

```
a = float(input("Enter first number: "))
b = float(input("Enter second number: "))

op = input("Enter operation (+, -, *, /,%): ")
if op == "+":
 print("Result:", a + b)
elif op == "-":
 print("Result:", a - b)
elif op == "*":
 print("Result:", a * b)
elif op == "/":
 print("Result:", a / b)
elif op == "%":
 print("Result:", a % b)
else:
 print("Invalid operation")
```

```
Enter first number:  16
Enter second number:  5
Enter operation (+, -, *, /,%):  %

Result: 1.0
```

34. Check if a bank account balance is sufficient for withdrawal

```
balance = float(input("Enter account balance: "))
withdraw = float(input("Enter withdrawal amount: "))
if balance >= withdraw:
 print("Withdrawal successful")
else:
 print("Insufficient funds")
```

```
Enter account balance:  25
Enter withdrawal amount:  5

Withdrawal successful
```

35. Implement a temperature converter □□□□

```
temp = float(input("Enter temperature: "))
unit = input("Enter unit (C/F): ").upper()
if unit == "C":
 print("Fahrenheit:", (temp * 9/5) + 32)
elif unit == "F":
 print("Celsius:", (temp - 32) * 5/9)
else:
 print("Invalid unit")
```

```
Enter temperature:  3
Enter unit (C/F):  C

Fahrenheit: 37.4
```

36. Check if a number lies within a range (50-100)

```
num = int(input("Enter a number: "))
if 50 <= num <= 100:
 print("Within range")
else:
 print("Out of range")
```

```
Enter a number:  85

Within range
```

37. Determine if a year is a century year

```
year = int(input("Enter a year: "))
if year % 100 == 0:
 print("Century Year")
else:
 print("Not a Century Year")
```

```
Enter a year:  2000

Century Year
```

38. Check if a number is a power of 2

```
num = int(input("Enter a number: "))
if num > 0 and (num & (num - 1)) == 0:
 print("Power of 2")
else:
 print("Not a Power of 2")
```

```
Enter a number:  1024

Power of 2
```

39. Determine how many days a month has □□□□□

```
month = int(input("Enter month (1-12): "))
year = int(input("Enter year: "))
days = [31, 28 + (1 if (year % 4 == 0 and year % 100 != 0) or (year %
400 == 0) else 0), 31,
30, 31, 30, 31, 31, 30, 31, 30, 31]
print("Days:", days[month - 1])
```

```
Enter month (1-12):  2
Enter year:  2002

Days: 28
```

40. Validate a password □□□□□

```
import re

password = input("Enter password: ")
```

```
if len(password) >= 8 and re.search(r"[A-Za-z]", password) and
re.search(r"\d", password):
 print("Valid Password")
else:
 print("Invalid Password")
# import re is used to bring in Python's Regular Expressions module,
    #which allows you to work with patterns in text – like matching,
searching, replacing, or splitting strings based on rules.

Enter password:  mahesh12345

Valid Password
```

This `if` condition checks for:

1. `len(password) >= 8` – Password must be at least 8 characters long.
2. `re.search(r"[A-Za-z]", password)` – Password must contain at least one letter (uppercase or lowercase).
3. `re.search(r"\d", password)` – Password must contain at least one digit.

HARD LEVEL

41. Implement a ticket pricing system

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

if age < 5:
 print("Ticket Price: Free")
elif age >= 60:
 print("Ticket Price: ₹50")
else:
 print("Ticket Price: ₹100")

Enter age:  30

Ticket Price: ₹100
```

42. Check if three numbers form a Pythagorean triplet □□□□□

```
a, b, c = sorted(map(int, input("Enter three numbers: ").split()))

if a**2 + b**2 == c**2:
 print("Pythagorean Triplet")
else:
 print("Not a Pythagorean Triplet")
```

```
Enter three numbers:  -3 4 5

Pythagorean Triplet
```

A Pythagorean triplet is a set of three positive integers (a, b, c) such that:

$$a^2 + b^2 = c^2$$

Here's a list of the smallest Pythagorean triplets (with $a < b < c$):

a	b	c
3	4	5
5	12	13
6	8	10
7	24	25
8	15	17
9	12	15
9	40	41
10	24	26
11	60	61
12	16	20
12	35	37
13	84	85
14	48	50
15	20	25
15	36	39
16	30	34

Sorted Pythagorean Triples With Hypotenuse Lengths Less Than 100

Sorted by Length of One Leg

3	4	5	**20**	15	25	**36**	15	39	**56**	33	65
4	3	5	**20**	21	29	**36**	27	45	**56**	42	70
5	12	13	**20**	48	52	**36**	48	60	**57**	76	95
6	8	10	**21**	20	29	**36**	77	85	**60**	11	61
7	24	25	**21**	28	35	**39**	52	65	**60**	25	65
8	6	10	**21**	72	75	**39**	80	89	**60**	32	68
8	15	17	**24**	7	25	**40**	9	41	**60**	45	75
9	12	15	**24**	10	26	**40**	30	50	**60**	63	87
9	40	41	**24**	18	30	**40**	42	58	**63**	16	65
10	24	26	**24**	32	40	**40**	75	85	**63**	60	87
11	60	61	**24**	45	51	**42**	40	58	**64**	48	80
12	5	13	**24**	70	74	**42**	56	70	**65**	72	97
12	9	15	**25**	60	65	**44**	33	55	**68**	51	85
12	16	20	**27**	36	45	**45**	24	51	**70**	24	74
12	35	37	**28**	21	35	**45**	28	53	**72**	21	75
13	84	85	**28**	45	53	**45**	60	75	**72**	30	78
14	48	50	**30**	16	34	**48**	14	50	**72**	54	90
15	8	17	**30**	40	50	**48**	20	52	**72**	65	97
15	20	25	**30**	72	78	**48**	36	60	**75**	40	85
15	36	39	**32**	24	40	**48**	55	73	**76**	57	95
16	12	20	**32**	60	68	**48**	64	80	**77**	36	85
16	30	34	**33**	44	55	**51**	68	85	**80**	18	82
16	63	65	**33**	56	65	**52**	39	65	**80**	39	89
18	24	30	**35**	12	37	**54**	72	90	**84**	13	85
18	80	82	**35**	84	91	**55**	48	73	**84**	35	91

Sorted by Hypotenuse

3	4	**5**	36	48	**60**						
6	8	**10**	11	60	**61**						
5	12	**13**	25	60	**65**						
9	12	**15**	39	52	**65**						
8	15	**17**	16	63	**65**						
12	16	**20**	33	56	**65**						
7	24	**25**	32	60	**68**						
15	20	**25**	42	56	**70**						
10	24	**26**	48	55	**73**						
20	21	**29**	24	70	**74**						
18	24	**30**	21	72	**75**						
16	30	**34**	45	60	**75**						
21	28	**35**	30	72	**78**						
12	35	**37**	48	64	**80**						
15	36	**39**	18	80	**82**						
24	32	**40**	13	84	**85**						
9	40	**41**	36	77	**85**						
27	36	**45**	40	75	**85**						
14	48	**50**	51	68	**85**						
30	40	**50**	60	63	**87**						
24	45	**51**	39	80	**89**						
20	48	**52**	54	72	**90**						
28	45	**53**	35	84	**91**						
33	44	**55**	57	76	**95**						
40	42	**58**	65	72	**97**						

✅ Can negative integers form a Pythagorean triplet?

Yes — algebraically, negative numbers can be part of a Pythagorean triplet because squaring a negative number makes it positive.

📌 For example:

Let's take $a = -3$, $b = 4$, and $c = 5$:

$$(-3)^2 + 4^2 = 9 + 16 = 25 = 5^2$$

✅ So $(-3, 4, 5)$ is also a valid Pythagorean triplet.

🧠 General Rule:

A triple (a, b, c) (where any of them could be negative) is still valid if:

$$a^2 + b^2 = c^2$$

So these are all valid:

- $(-3, 4, 5)$
- $(3, -4, 5)$
- $(-3, -4, 5)$

⚠️ But in classical number theory:

Pythagorean triplets are usually defined as positive integers only.

Would you like a Python program to generate such "signed" triplets too?

43. Convert a Roman numeral to an integer□□□□

```
def roman_to_int(s):
 roman = {'I': 1, 'V': 5, 'X': 10, 'L': 50, 'C': 100, 'D': 500, 'M':
1000}
 total = 0
 for i in range(len(s)):
 if i > 0 and roman[s[i]] > roman[s[i - 1]]:
 total += roman[s[i]] - 2 * roman[s[i - 1]]
 else:
 total += roman[s[i]]
 return total
num = input("Enter Roman numeral: ").upper()
print("Integer:", roman_to_int(num))
```

44. Determine zodiac sign based on birth date □□□□

```
month = int(input("Enter birth month (1-12): "))
day = int(input("Enter birth day: "))
zodiac = [("Capricorn", 20), ("Aquarius", 19), ("Pisces", 20),
("Aries", 20),
 ("Taurus", 21), ("Gemini", 21), ("Cancer", 22), ("Leo", 22),
 ("Virgo", 22), ("Libra", 23), ("Scorpio", 23), ("Sagittarius", 22),
("Capricorn", 31)]

sign = zodiac[month - 1][0] if day <= zodiac[month - 1][1] else
zodiac[month][0]
print("Zodiac Sign:", sign)
```

```
Enter birth month (1-12):  12
Enter birth day:  2

Zodiac Sign: Sagittarius
```

45. Check if a number is a Harshad number □□□□□

A Harshad number, also known as a Niven number, is a positive integer that is divisible by the sum of its digits. For example, 18 is a Harshad number because the sum of its digits (1 + 8 = 9) divides 18 evenly.

- 12 is a Harshad number because 1 + 2 = 3, and 12 is divisible by 3.
- 27 is a Harshad number because 2 + 7 = 9, and 27 is divisible by 9. example: [1 2 3 4 5 6 7 8 9 10 12 18 20 21 24 27 30 36 40 42 45 48 50 54 60 63 70 72 80 81 84 90 100]

```
num = int(input("Enter a number: "))
sum_digits = sum(int(digit) for digit in str(num))
if num % sum_digits == 0:
 print("Harshad Number")
else:
 print("Not a Harshad Number")
```

```
Enter a number:  100

Harshad Number
```

46. Validate an email format □□□□

```
import re
email = input("Enter email: ")
if re.match(r"^[\w\.-]+@[\w\.-]+\.(com|org|net|edu)$", email):
#is using regular expressions (re.match) in Python to validate an
email address.
     print("Valid Email")
else:
 print("Invalid Email")
```

```
Enter email:  mahes@gmai.com

Valid Email
```

47. Check if a knight move in chess is valid □□□□□

```
x1, y1 = map(int, input("Enter current position (x y): ").split())
x2, y2 = map(int, input("Enter new position (x y): ").split())
if (abs(x1 - x2), abs(y1 - y2)) in [(2, 1), (1, 2)]:
 print("Valid Knight Move")
else:
 print("Invalid Move")
```

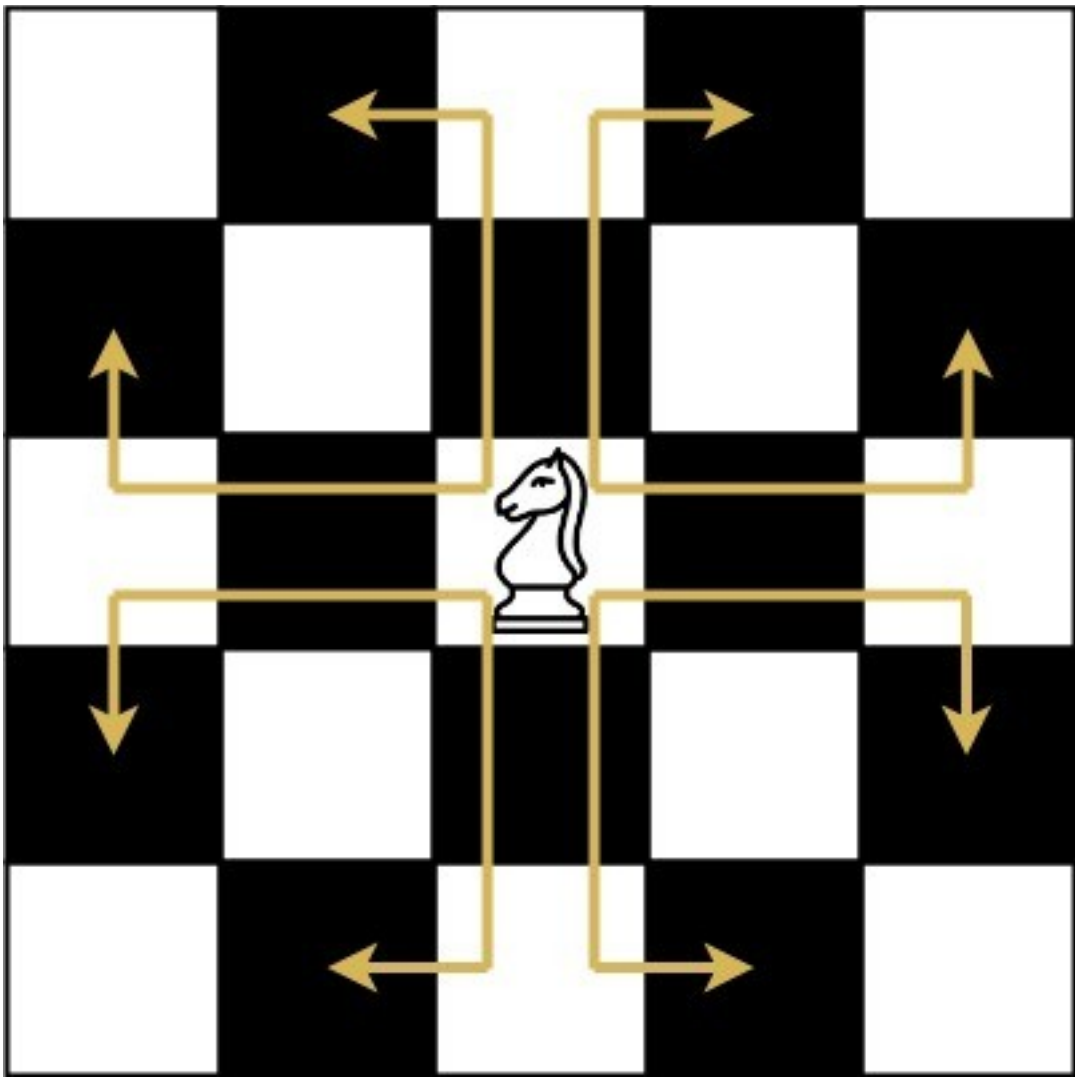
```
Enter current position (x y):  4 4
Enter new position (x y):  6 5

Valid Knight Move
```

✅ Example Valid Inputs:

- From `(1, 1)` to `(2, 3)` → `abs(1 - 2), abs(1 - 3)` → `(1, 2)` ✅
- From `(4, 4)` to `(6, 5)` → `(2, 1)` ✅

❌ Invalid Input:

- From `(1, 1)` to `(3, 3)` → `(2, 2)` ❌ Not a knight move



48. Implement a loan eligibility checker □□□□

```
income = int(input("Enter monthly income: "))
credit_score = int(input("Enter credit score: "))
employed = input("Are you employed? (yes/no): ").lower()

if income >= 25000 and credit_score >= 700 and employed == "yes":
 print("Loan Approved")
else:
 print("Loan Denied")
```

```
Enter monthly income:  8000
Enter credit score:  76
Are you employed? (yes/no):  yes
```

```
Loan Denied
```

49. Implement a rock-paper-scissors game □□□□

```
import random

choices = ["rock", "paper", "scissors"]
user = input("Enter rock, paper, or scissors: ").lower()
computer = random.choice(choices)

print("Computer chose:", computer)

if user == computer:
    print("It's a tie!")
elif (user == "rock" and computer == "scissors") or \
     (user == "scissors" and computer == "paper") or \
     (user == "paper" and computer == "rock"):
    print("You win!")
else:
    print("You lose!")
```

```
Enter rock, paper, or scissors:  paper

Computer chose: paper
It's a tie!
```

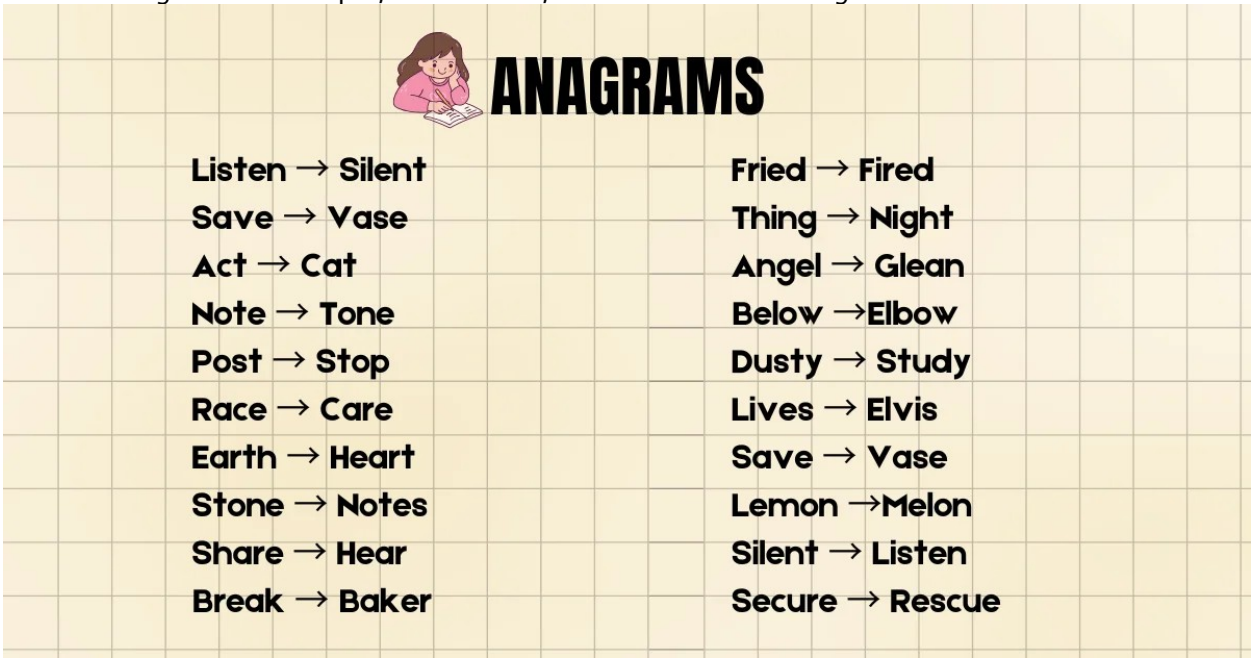
50. Find the day of the week for a given date (without built-in functions □□□□

```
 def day_of_week(d, m, y):
    if m < 3:
        m += 12
        y -= 1
    k = y % 100
    j = y // 100
    day = (d + (13 * (m + 1)) // 5 + k + (k // 4) + (j // 4) - 2 * j)
% 7
    days = ["Saturday", "Sunday", "Monday", "Tuesday", "Wednesday",
"Thursday", "Friday"]
    return days[day]

# Get user input
day, month, year = map(int, input("Enter date (DD MM YYYY):
").split())
print("Day of the Week:", day_of_week(day, month, year))
```

```
Enter date (DD MM YYYY):  8 1 2025

Day of the Week: Wednesday
```

Anagram Number

An Anagram Number is a number that can be rearranged to form another number using the exact same digits. For example, 123 and 321, or 112 and 211 are anagrams




```python
# for number
def are_anagram_numbers(num1, num2):
    # Convert both numbers to strings
    str1 = str(num1)
    str2 = str(num2)

    # Sort the digits and compare
    return sorted(str1) == sorted(str2)

# Input two numbers
n1 = int(input("Enter first number: "))
n2 = int(input("Enter second number: "))

# Check if they are anagram numbers
if are_anagram_numbers(n1, n2):
    print("✅ The numbers are anagrams of each other.")
else:
    print("❌ The numbers are NOT anagrams of each other.")
```

```
Enter first number:  123
Enter second number:  321

 The numbers are anagrams of each other.
```

```python
# for number
def are_anagram_strings(str1, str2):
    # Remove spaces and convert to lowercase
    str1 = str1.replace(" ", "").lower()
    str2 = str2.replace(" ", "").lower()

    # Compare sorted characters
    return sorted(str1) == sorted(str2)

# Input two strings
s1 = input("Enter first string: ")
s2 = input("Enter second string: ")

# Check and print result
if are_anagram_strings(s1, s2):
    print(" The strings are anagrams.")
else:
    print(" The strings are NOT anagrams.")
```

```
Enter first string:  anna
Enter second string:  nana

 The strings are anagrams.
```

No.	Question	Description
1.	**Anagram Checker**	Check if two strings are anagrams (already done).
2.	**Palindrome Checker**	Check if a string reads the same forwards and backwards.
3.	**Reverse a String**	Print or return the reversed version of a string.
4.	**Count Vowels and Consonants**	Count the number of vowels/consonants in a string.
5.	**Remove Duplicates**	Remove duplicate characters from a string.
6.	**Find Frequency of Characters**	Count how many times each character appears.
7.	**Check Pangram**	Check if a string contains every letter of the alphabet at least once.

No.	Question	Description
18.	**Find First Non-Repeating Character**	Find the first character that doesn't repeat.
19.	**String Compression**	Compress a string like `'aaabb' → 'a3b2'`.
10.	**Check Isogram**	Check if a word has all unique characters (no repeats).
11.	**String Permutations**	Print all permutations of a given string.
12.	**Capitalize Each Word**	Convert first letter of each word to uppercase.
13.	**Check Rotation**	Check if one string is a rotation of another.
14.	**Valid Parentheses**	Check if the input string has balanced brackets.
15.	**Longest Word in Sentence**	Find the longest word from a sentence.
16.	**Word Count**	Count the number of words in a string.
17.	**Check Substring**	Determine if one string is a substring of another.
18.	**Replace Character**	Replace a character with another in a string.
19.	**Count Digits, Letters, Special Chars**	Classify and count characters in the string.
20.	**Check Palindromic Substrings**	Find/count palindromes within a longer string.

Top Technical Round Questions (Beginner to Intermediate)

String-Based Questions

1. Check if two strings are anagrams.

2. Check if a string is a palindrome.
3. Reverse a string without using built-in functions.
4. Count vowels and consonants in a string.
5. Remove duplicate characters from a string.
6. Check if a string is a pangram.
7. Print all permutations of a string.
8. Replace all spaces in a string with `%20`.
9. Compress a string like `"aaabbc"` to `"a3b2c1"`.
10. Find the first non-repeating character in a string.

Number-Based Questions

1. Check if a number is **prime**.
2. Check if a number is **Armstrong**.
3. Check if a number is **Harshad/Niven** (divisible by sum of its digits).
4. Reverse a number.
5. Check if a number is a palindrome.
6. Check if two numbers are anagrams.
7. Find factorial of a number (using loop and recursion).
8. Generate Fibonacci series.
9. Check if a number is perfect.
10. Count digits in a number without using `len()`.

Array-Based Questions

1. Find largest/smallest element in an array.
2. Reverse an array without using `[::-1]`.
3. Find the duplicate elements in an array.
4. Check if two arrays are equal (same elements, any order).
5. Merge two sorted arrays.
6. Move all zeros to the end of an array.
7. Remove duplicates from an array.
8. Find the second largest element.
9. Rotate an array left/right by `k` steps.
10. Find missing number in a sequence (e.g., 1 to n).

Pattern Printing Questions

1. Print right-angled triangle.
2. Print pyramid pattern.
3. Print number pyramid.
4. Print inverted triangle.
5. Print diamond pattern.

Logic & Algorithm-Based

1. FizzBuzz (print "Fizz" for multiples of 3, "Buzz" for 5, "FizzBuzz" for both).
2. Check if a string or number is a palindrome.
3. Check balanced parentheses.
4. Implement binary search.
5. Find GCD and LCM of two numbers.
6. Find length of longest word in a sentence.
7. Implement linear search.
8. Count frequency of characters or digits.
9. Sort an array (bubble/selection/insertion sort).
10. Detect if a number is in Fibonacci sequence.

```
# Input a string from the user
alphanumeric = input("Enter an alphanumeric string: ")

# Initialize two empty strings for letters and digits
letters = ""
digits = ""

# Loop through each character in the input string
for char in alphanumeric:
    if char.isalpha():
        letters += char
    elif char.isdigit():
        digits += char

# Print the results
print("Alphabets:", letters)
print("Digits:", digits)
```

```
Enter an alphanumeric string:  mahesh12345

Alphabets: mahesh
Digits: 12345
```