

Python Logic

1. Swap 2 variable without using 3rd variable. (Give minimum three solutions)
2. Accept the marks from the user and print the corresponding grade
 - a. Marks ≥ 75 grade A
 - b. Marks ≥ 55 grade B
 - c. Marks ≥ 35 grade C
 - d. Else fail
3. Accept a number from the user - if it is divisible by 3 print "Three" , if it is divisible by 7 print "Seven" and if it is divisible by both 3 & 7, print "Three - Seven".
4. Accept a number from the user and check if it is odd or even number (Hint: use % operator).
5. Accept a number from the user check if it is odd or even number (Do not use % operator).
6. Accept principal amount, rate of interest and years of investment then find the simple interest.
7. Problem: Given the length of four sides of a figure, determine whether they can be used to create a polygon and determine if that polygon is a square. A polygon can be created if no single side is greater than the sum of the other three sides. Display a ONE when the status (for polygon or square) is confirmed and ZERO when the status cannot be confirmed.
Example Execution #1:
Enter the length of the four sides: 4 3 2 9
Length of sides: 4, 3, 2, 9
Polygon status: 1
Square status: 0

Example Execution #2:
Enter the length of the four sides: 5 5 5 5
Length of sides: 5, 5, 5, 5
Polygon status: 1
Square status: 1

Example Execution #3:
Enter the length of the four sides: 84 25 20 15
Length of sides: 84, 25, 20, 15
Polygon status: 0
Square status: 0
8. Accept 10 numbers from the user and calculate their sum. (Do not use array).
9. Accept a number from the user and find the factorial of the number.
10. Accept 10 numbers from the user and count how many are positive, negative or zero.
11. Accept a number from the user and calculate the sum of the digits of the number.

12. Accept a number from the user and reverse it. Accept a number from the user and check if it is palindrome number or not.
13. Accept a number from the user and print a table for that number.
14. Accept a number from the user check if it is special number or not?
Example: 145
 $1! = 1$
 $4! = 1*2*3*4$ $5! = 1*2*3*4*5$
Sum of the factorials is $(1+24+120=145)$
15. Accept a number from the user and check if it is an Armstrong number or not?
Example: 153
1 cube = 1
5 cube = 125
3 cube = 27
Sum of the cubes is $(1+27+125=153)$
16. Go on accepting numbers from the user till the user enters 0 and calculate the sum of these numbers.
17. Accept a number from the user and print the next 5 numbers.
18. Accept a number from the user and print those many numbers after the number.
19. Accept a start and end range from the user and print all even number between them. (Give minimum two solutions).
20. Accept start and end range from the user and print all odd numbers between them. (Give minimum two solutions).
21. Accept a number from the user and print the Fibonacci series till that number.
22. Accept a number and a single digit from the user. Then check the number of occurrences of that digit in the number.
23. Accept 10 numbers from the user and print the highest number.
24. Accept 10 numbers from the user and print the lowest number.
25. Accept 10 numbers from the user and print the highest and the 2nd highest number.
26. Write a program to print every integer between 1 and n divisible by m. Also report whether the number that is divisible by m is even or odd.
27. Print all special numbers between 1 to 200000.
28. Print all Armstrong number between 1 to 200000.
29. Accept a number from the user and check if it is a prime number or not.
30. Print all prime number between 51 to 100.

31. Print all twin prime numbers between 1 to 20.

32. Write programs to print the following Patterns:

1 12 123	1 22 333	1 2 2 3 3 3
1 23 456	1 12 123	1 21 321
1 121 12321	1 121 12321 121 1	32123 212 3
1 222 33333	0 101 21012	zyxyz zyz z
a ab abc	* ** *** ****	** * * **
--*-- -*--* *---*	1 121 12121	321 32 3

33. Accept a number from the user and print that many prime numbers after the number.

Example: Input: 3 Output: 5,7,11

34. Accept a number from the user and find the factorial of the number using recursion.

35. Convert the given number to Binary using recursion.

36. Convert the given number to Hexadecimal using recursion.

37. Accept a number from the user and print the Fibonacci series till that number using recursion.

38. Accept two numbers from the user and find the HCF and LCM of the numbers using recursion.

39. Accept 5 numbers in a list, accept a number from the user and check if given number is there in the list or not.

40. Accept 5 numbers in a list and sort it (bubble sort).

41. Accept 5 numbers in a list and sort it (selection sort).

42. Accept 10 numbers in a list and sort it (insertion sort).
43. Accept 5 numbers in a list and sort it (Merge sort).
44. Accept 5 numbers in a list, accept a number from the user and check if given number is there in the list or not. (use binary search).
45. Accept data in two 3*3 matrix and calculate the sum of the matrices.
46. Accept data in two 3*3 matrix and calculate the product of the matrices.
47. Accept data in a 4*4 matrix and transpose it.
48. Accept data in a 4*4 matrix and check if it is a magic square or not.
49. Accept data in 3*3 matrix and print row wise and column wise total.
50. Write a sort algorithm for following array with all zero first.
Input string: [0,3,7,1,22,3,0,6,8,9,0] Output string: [0.0.0,1,3,3, 6 ,7 ,8 .9 .22]
50. Accept a string from the user, accept a character from the user and count the occurrence of the same.
51. Accept a string and check if it is a palindrome or not. Example: "NITIN"
52. Accept a string and reverse it.
53. Accept a sentence from the user and count number of words.
54. Accept a sentence from the user and reverse the words in the sentence.
55. Accept two strings from the user and append 1st one with the second one.
56. Accept two string and check both are same or not.
57. Accept 5 names from the user and print them in ascending order. Accept a name from the user and check that name is there in array or not.
58. Accept a string accept a character and delete all occurrence of that character.
59. Take a paragraph as an input from user. If user enter more than 500 char, it should show warning.
Following string will be given. Vehicle: bus, bike, car, rickshaw. Subject: maths, physics, chemistry, social study.
lang: c, java, vb, ruby
Find out the occurrence of all of the above and print them in sorted order.
eg. input=Dafsafsdjavajavacrubyc#mathssocialstudy
output: c 2
 c# 1
 java 2

maths 1

ruby 1

60. Define a structure called cricket that will describe the following information

- Player name
- Team name
- Batting average

Using cricket, declare an array (player) with 10 elements and write a program to read the information about all the 10 players and print a team wise list containing names of players with their batting average.

61. Define a structure called student that will describe the following information

- Name
- Div
- %obtain

Using student, declare an array mark sheet with 10 elements and write a program to read the information about all the 10 students and print a division wise list containing names of student with their % obtain

62. You have to write a program to count the number of words in the input. The program should read in the input text till end_of_file (EOF) and output the number of words found. A word can be taken to be a sequence of alphanumeric characters terminated by a space or by a newline. Assume that there will not be any characters other than alphanumeric (a-z, A-Z, 0-9) and white spaces (blank, tabs and newlines) in the input.

63. Write a program that reads in a text stream from standard input and uses a stack to determine whether its parentheses are properly balanced. For example, your program should print true for `[]{}{[]()}{}` and false for `[]()`. Hint : Use a stack.

64. Write a program that takes from standard input an expression without left parentheses and prints the equivalent infix expression with the parentheses inserted. For example, given the input

`1 + 2) * 3 - 4) * 5 - 6))`

Your program should print

`((1 + 2) * ((3 - 4) * (5 - 6))`

65. Write a program that converts an arithmetic expression from infix to postfix

66. Write an algorithm to detect a loop in a singly linked list of unknown size?

67. Write a program to maintain a singly linked list having the following functions:

- | | | | | |
|-------------------------|------------------------|------------------|----------------|----------------|
| a. Creation of the list | b. Displaying the list | c. Sort the list | d. Insert node | e. delete node |
|-------------------------|------------------------|------------------|----------------|----------------|

68. Design an algorithm for an Elevator(not the code)

69. Accept date of birth(DD MM YY) and current month. Calculate the age in number of years and months.

70. You are given an array of size N. The array is formed using a tree rooted at node 1.

Array generation code

generate (node)

```
{
    if (node == 1)
        insert node into the array
        visited [node]=true;
    for(all nodes in the adjacency list of node)
    {
        child = node in adjacency list
        if (child is not visited)
        {
            insert child into the array
            generate (child)
            insert node into array
        }
    }
}
```

You have to obtain the tree using the array and print the number of nodes in the subtree of every node.

71. Unique Subsequence : You are given randomly generated string of characters ranging from a to z. You have to tell no of unique subsequences that can be formed equal to following string.

“a b c d e f g h i j k l m n o p q r s t u v w x y z”

Note : A subsequence is unique when at least a single character has its index i.e. different from all other subsequences.

INPUT	OUTPUT
3	
27	
a b c d e f g h i j k l m n n o p q r s t u v w x y z z	2
Abz	0
a b c d e f g h i j k l m n n o p q r s t u v v w x y y z z	4

72. Consider N walls, each of unit width, situated next to each other. You have to select any two walls i and j ($1 \leq i, j \leq N$) such that if you break all walls except i and j and fill spaces between them with water, then the amount of water stored is maximum. Write a program to find the maximum amount of water that can be accumulated between the walls.

73. Write a crawler. You are given Directory (ABC) structure. How will you iterate and give all the children(B,C,D,F)

Input:

A->B->C

B->D

C->F

Example:

Google-> Gmail-> YouTube

Gmail-> Google Drive-> Google Photo

YouTube->Facebook

74. Forgotten Report

There are N cities in a state. You start your ride from first city. You have to visit all other cities exactly once and finally return to your origin city. After visiting each city, you collect analysis report. But when you reached last visited city, you remembered that you did not collect report from city K. So, now you decide to first collect report from city K and then return to your home city.

Given distances between each pair of cities, you are required to find shortest possible distance for your whole journey.

INPUT :

- 1) input begins with T (no. of test cases)
- 2) Contains K city No. Where you forget to collect report.
- 3) Contains N (no. of cities)
- 4) There are N lines. Ith line have exactly N numbers denoting distance from city I to all 'N' cities.

OUTPUT :

For each test case print Minimum distance of total journey.

Constraints	Sample Input
$1 < T < 10$	1
$1 < N < 18$	2
$1 < K < N$	4
$0 < \text{dist}(i, j) < 100$	0 1 1 5 6
	2 0 7 3

75. You are a bank owner, how will you decide in which locations should you deploy ATM Machine?

76. If you are a car manufacturer, which development methodology would you choose – Agile/ Waterfall.