

Practice Problems in Python

BSDS, 1st Year 2025–2026

Date: 11 August, 2025

PRACTICING INSTRUCTIONS

1. Write down the best possible implementation of the problems listed below.
2. Take care of the error cases too.
3. After you finish the coding, look at the model answers at the end and understand.

NOTE: The programs are to be written in Python and should be well commented. All programs should take the required inputs from standard input buffer and print the desired outputs to the standard output buffer, until otherwise stated.

Q1. (EASY, ChatGPT 5.0 Compatible) Write a program to print the following pattern given the line number as user input.

```
*****
  *
 *
*
 *
*
*****
```

Q2. (EASY, ChatGPT 5.0 Compatible) Given an integer, write a program to verify whether the integer has the same number of even digits and odd digits or not.

Q3. (EASY, ChatGPT 5.0 Compatible) Write a program that automatically converts English text to Morse code.

Q4. (EASY, ChatGPT 5.0 Compatible) Given the first $n - 1$ elements of a permutation of the first n natural numbers, write a program to find out the last element. Note that n is unknown but in the form 2^k .

Q5. (EASY, ChatGPT 5.0 Compatible) The Pearson correlation coefficient (ρ_{xy}) between two sets of real values $\{x_1, x_2, \dots, x_n\}$ and $\{y_1, y_2, \dots, y_n\}$, corresponding to n samples of two sets of variables, is calculated as follows:

$$\rho_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}.$$

Write a program to calculate the Pearson correlation coefficient from two given sets of real values.

- Q6. (EASY, ChatGPT 5.0 Compatible) Let us define the **palindrome factor** of a string as the number of characters that is to be substituted (i.e., the edit distance) to obtain its reverse string. For example, **palindrome factor** of the string “malay” is 2 because “malay” and “yalam” have an edit distance of 2. Write a program that will take a string as input and show its **palindrome factor** as the output.
- Q7. (EASY, ChatGPT 5.0 Incompatible) A number is called a *Perfect Digital Invariant* (PDI) of order p if the sum of its own digits each raised to the power p is equal to the number. For example $1 = 1^2$ is a PDI of order 2, $153 = 1^3 + 5^3 + 3^3$ is a PDI of order 3, and $9474 = 9^4 + 4^4 + 7^4 + 4^4$ is a PDI of order 4. Write a program to verify whether a given number is PDI or not for any arbitrary p .
- Q8. (EASY, ChatGPT 5.0 Incompatible) X and Y are 1km apart. They start running toward each other at 12 and 18 kmph, respectively. As they start, a bird sitting on the cap of X, starts flying toward Y, touches Y and returns back to X and so on, till they meet. Write a program to calculate the distance traveled by the bird, if its speed in kmph is given as input.
- Q9. (MEDIUM, ChatGPT 5.0 Compatible) Given a set of n numbers and an index $i \in \{1, \dots, n\}$ as user input, write a program to efficiently find out the nearest larger value for the number at index i .
- Q10. (MEDIUM, ChatGPT 5.0 Compatible) Given the square matrix A and the nonnegative integer n , write a program that can recursively compute A^n involving minimal multiplications.
- Q11. (MEDIUM, ChatGPT 5.0 Compatible) Write a program to print the following pattern given the line number as user input. Note that the numbers starting from all the four corners are in the ascending order following the layers toward the center.
- ```

1 2 3 2 1
2 2 3 2 2
3 3 3 3 3
2 2 3 2 2
1 2 3 2 1

```
- Q12. (MEDIUM, ChatGPT 5.0 Compatible) Given a large integer  $n$  as user input, write an efficient program to return its prime factors.
- Q13. (MEDIUM, ChatGPT 5.0 Compatible) A number is called a **Vampire Number** if one can take its digits, re-arrange them into 2 new numbers and multiply them to get the original number back. Write a program to verify whether a given number is a **Vampire Number**.
- Q14. (MEDIUM, ChatGPT 5.0 Compatible) Given a pair of strings as user input, write a program that returns the shortest sequence of character insertions and deletions that turn one string into the other.

- Q15. (MEDIUM, ChatGPT 5.0 Compatible) Write a program to find out the longest palindromic substring of a string given as user input.
- Q16. (MEDIUM, ChatGPT 5.0 Compatible) There is a river with a boat. Some people standing on a side of the river are to be transferred to the other side. The boat can carry at most two persons costing a fare proportional to the sum of the weights of the passengers. Given a set of weights of the passengers, write a program to find out the minimum cost of transferring all of them. Note that the boat cannot row empty.
- Q17. (MEDIUM, ChatGPT 5.0 Incompatible) Write a program to take two extremely large numbers (that does not fit into the RAM) as inputs and return the summation of tens digit and units digit of their product.
- Q18. (HARD, ChatGPT 5.0 Compatible) Given a multiset of positive integers, write a program to partition it (if possible) into two subsets of equal sum.
- Q19. (HARD, ChatGPT 5.0 Incompatible) Let us define the **SQUEEZE** operation on a string that decreases the string of occurrences of all of its characters same number of times retaining their ordering in the original string. E.g., if a number ‘aaabbbbaaa’ is **SQUEEZED** by a factor of 3 then it will turn into aba. Write a program that will take a string and show the maximum possible **SQUEEZED** string as the output. Note that the **SQUEEZE** operation is valid if and only if it is applicable to all the segments (series of characters) of a particular string.
- Q20. (OPEN, ChatGPT 5.0 Incompatible) Andrica’s conjecture (named after Romanian mathematician Dorin Andrica (es)) is a conjecture regarding the gaps between prime numbers. The conjecture states that the inequality

$$\sqrt{p_{n+1}} - \sqrt{p_n} < 1$$

holds for all  $n$  where  $p_n$  is the  $n^{th}$  prime number.

Write an efficient program to either verify its correctness quickly for any arbitrary  $n$  or find out a counterexample.

## Model Answers:

### Q1. GENERAL APPROACH

```
line = int (input('Enter an number: '))
for i in range(line):
 for j in range(line):
 if i == 0 or i == line - 1:
 print("*", end = "")
 elif j == line - i - 1:
 print("*", end = "")
 else:
 print(" ", end = "")
 print()
```

### BETTER APPROACH

```
line = int(input('Enter an number: '))
print("*" * line)
for i in range(line - 2):
 print(" " * (line - i - 2), "*", sep = "")
print("*" * line)
```

### Q2. GENERAL APPROACH

```
num = input('Enter an integer: ')
even = '02468'
countE, countO = 0, 0
for d in num:
 if d in even:
 countE = countE + 1
 else:
 countO = countO + 1
if countE == countO:
 print('SAME')
else:
 print('DIFFERENT')
```

### BETTER APPROACH

```
num = input('Enter an integer: ')
even = "02468"
countE = sum(num.count(d) for d in even)
print('DIFFERENT' if len(num)-(countE<<1) else 'SAME')
```

### Q3. GENERAL APPROACH

```

MORSE_CODE_DICT = {
 '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': '--..', '0': '-----', '1': '.----', '2': '..---', '3': '...--',
 '4': '....-', '5': '.....', '6': '-....', '7': '--...',
 '8': '---..', '9': '----.', ',': '--...', '.': '.-.-.-', '?': '..-.-.',
 '/': '-.-.-', '-': '-....-', '(': '-.-.-', ')': '-.-.-.-', ' ': '/'
}

def text_to_morse(text):
 text = text.upper()
 morse = []
 for ch in text:
 if ch in MORSE_CODE_DICT:
 morse.append(MORSE_CODE_DICT[ch])
 else:
 morse.append('')
 return ' '.join(morse)

english = input('Enter an English text: ')
morse = text_to_morse(english)
print('The Morse code is: ', morse)

```