





# Python Practice Questions







1, A shop gives a discount if a customer buys more than 3 items.  
Write a program that asks the user for the number of items they want  
to buy and prints:

"Discount applied" if items > 3  
"No discount" otherwise

2, You are given a list of product prices:  
prices = [120, 45, 300, 85, 150]  
Write a function get\_expensive\_products(prices) that returns a new  
list containing only the prices greater than 100.





3, Write a script that logs user activity.

When the program runs

Write "User logged in" to a file called log.txt.

Then read the file and print the full log history.

4, You are building a small student grade system.

Write a function:

```
get_grade(student_grades, student_name)
```


It should:

Try to return the student's grade from a dictionary

If the student does not exist, catch the exception and return:

"Student not found in the system"





5, You receive a file called sales.txt where each line should contain a sales number.

Example:





200

450

abc

700



- 
- 
- 6, Write a program that:
- a, Reads every line in sales.txt
  - b, Converts valid lines into integers
  - c, Skips invalid lines (like "abc") using exception handling
  - d, Stores the valid numbers in a list
  - e, Calculates and prints the total sales
- 
- 



7. Given an integer  $n$ , return a string array `answer` (**1-indexed**) where:

- `answer[i] == "FizzBuzz"` if  $i$  is divisible by 3 and 5.
- `answer[i] == "Fizz"` if  $i$  is divisible by 3.
- `answer[i] == "Buzz"` if  $i$  is divisible by 5.
- `answer[i] == i` (as a string) if none of the above conditions are true.

**Example 1:**

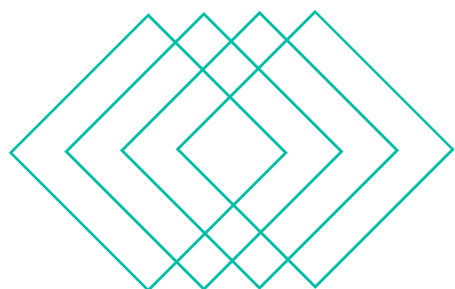
**Input:**  $n = 3$

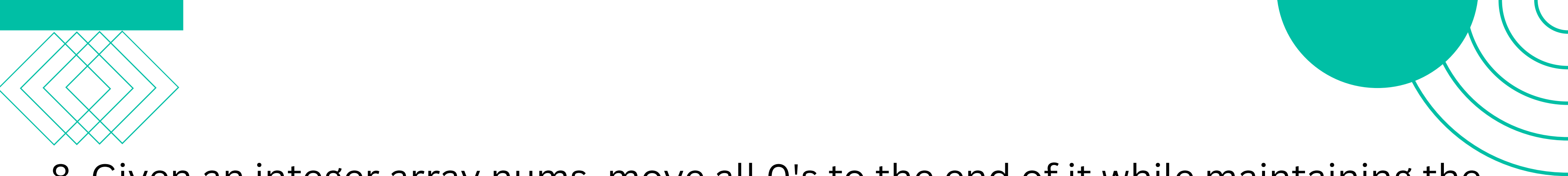
**Output:** `["1","2","Fizz"]`

**Example 2:**

**Input:**  $n = 5$

**Output:** `["1","2","Fizz","4","Buzz"]`





8. Given an integer array `nums`, move all 0's to the end of it while maintaining the relative order of the non-zero elements.

**Note** that you must do this in-place without making a copy of the array.

**Example 1:**

**Input:** `nums = [0,1,0,3,12]`

**Output:** `[1,3,12,0,0]`

**Example 2:**

**Input:** `nums = [0]`

**Output:** `[0]`





9. Given an array of integers `nums` and an integer `target`, return indices of the two numbers such that they add up to `target`. You may assume that each input would have **exactly one solution**, and you may not use the same element twice.

You can return the answer in any order.

**Example 1:**

Input: `nums = [2,7,11,15]`, `target = 9`

Output: `[0,1]`

Explanation: Because `nums[0] + nums[1] == 9`, we return `[0, 1]`.

**Example 2:**

**Input:** `nums = [3,2,4]`, `target = 6`

**Output:** `[1,2]`







10. Given an integer  $x$ , return true if  $x$  is a palindrome, and false otherwise.

**Example 1:**

**Input:**  $x = 121$

**Output:** true

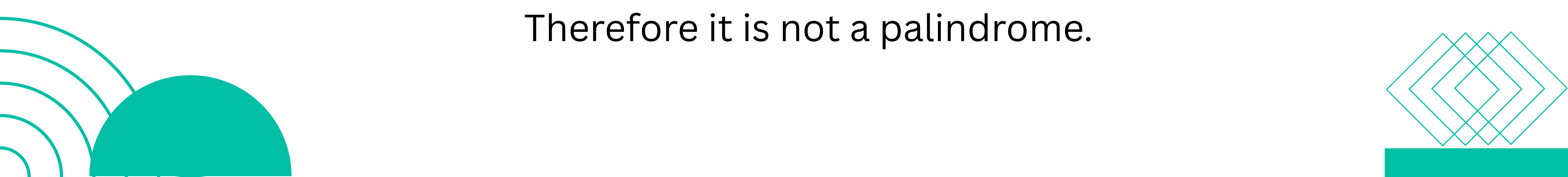
**Explanation:** 121 reads as 121 from left to right and from right to left.


**Example 2:**

**Input:**  $x = -121$

**Output:** false

**Explanation:** From left to right, it reads -121. From right to left, it becomes 121-. Therefore it is not a palindrome.





11. Given a non-negative integer  $x$ , return the square root of  $x$  rounded down to the nearest integer. The returned integer should be non-negative as well.

You must not use any built-in exponent function or operator.

- For example, do not use  $x^{**} 0.5$

**Example 2:**

**Input:**  $x = 8$

**Output:** 2

**Explanation:** The square root of 8 is 2.82842..., and since we round it down to the nearest integer, 2 is returned.

**Example 1:**

**Input:**  $x = 4$

**Output:** 2

**Explanation:** The square root of 4 is 2, so we return 2.





12. A file named `numbers.txt` contains one number per line.

examples `numbers.txt`:

10

20

30

abc

40

Write a program that:

- Reads each line
- Converts valid numbers to integers
- Ignores invalid lines (use try/except)
- Prints the sum of all valid integers

Expected output:

Sum = 100 (because "abc" is skipped)





## 13. Read File and Convert All Text to Uppercase

Write a program that:

- Reads a file
- Converts all text to uppercase
- Displays it on the screen
- Uses try/except to handle missing file errors

**Input:** message.txt:

Hello World  
Python is fun

**output:**

HELLO WORLD  
PYTHON IS FUN





14. You are given a dictionary storing student scores:

```
scores = {"John": 85, "Sara": 92, "Fraol": 78}
```

Write a program that:

1. Asks the user to enter a student name.
2. Tries to print the student's score from the dictionary.
3. If the key does not exist, catch the exception and print:  
"Student not found!"

**Example:**

**Input:** Mark

**Output:** Student not found!





15. Write a program that:



- Takes a sentence as input
- Splits it into words
- Stores the frequency of each word in a dictionary
- Prints the dictionary

**Example:**

**Input:** "python is fun and python is powerful"

**Output:** {"python": 2, "is": 2, "fun": 1, "and": 1, "powerful": 1}





16. Create a new dictionary where the keys become values and values become keys.

**Example:**

**Input:** grades = {"John": "A", "Sara": "B", "Musa": "A"}

**output:** {"A": ["John", "Musa"], "B": ["Sara"]}

