

## Editorial-Mock: End Module A Quiz

### Question 1:

A company is analyzing two algorithms, Algorithm A and Algorithm B, to process customer orders.

- Algorithm A has a time complexity of  $O(n)$  and space complexity of  $O(1)$ .
- Algorithm B has a time complexity of  $O(n^2)$  and space complexity of  $O(n)$ .

### Options:

1. Algorithm A is more time-efficient than Algorithm B for large input sizes.
2. Algorithm B uses more memory than Algorithm A for large input sizes.
3. Algorithm A is faster for small input sizes  $n < 10$
4. Algorithm B's time complexity indicates it will always perform better than Algorithm A.

### Answer and Explanation :

- Correct options: 1, 2
- Algorithm A's  $O(n)$  time complexity is more efficient for large input sizes compared to Algorithm B's  $O(n^2)$
- Algorithm B's space complexity of  $O(n)$  means it uses more memory than Algorithm A's  $O(1)$ .
- The performance of Algorithm A versus Algorithm B for small inputs depends on constant factors, so statement 3 is not guaranteed.
- Statement 4 is incorrect as  $O(n^2)$  is less efficient than  $O(n)$ .

### Question 2:

A software engineer has implemented a function that calculates the sum of all elements in an array of size  $n$ . The function uses a single loop and an integer variable to store the sum. What are the time and space complexities of this function?

### Options:

1.  $O(n)$ ,  $O(1)$
2.  $O(n^2)$ ,  $O(n)$
3.  $O(1)$ ,  $O(1)$
4.  $O(n)$ ,  $O(n)$

### Answer and Explanation:

- Correct option: 1

- The function iterates through all  $n$  elements once, making the time complexity  $O(n)$ .
- It uses a single integer variable to store the sum, so the space complexity is  $O(1)$ .

### Question 3:

A software engineer at a finance firm writes a program to read a file and count the number of lines in it. The program should handle cases where the file does not exist or there are issues reading the file. The code snippet is as follows:

try:

```
filename = input("Enter the file name: ")
```

```
with open(filename, "r") as file:
```

```
    lines = file.readlines()
```

```
    print(f"The file has {len(lines)} lines.")
```

```
except FileNotFoundError:
```

```
    print("Error: The file does not exist.")
```

```
except PermissionError:
```

```
    print("Error: You do not have permission to read the file.")
```

```
else:
```

```
    print("File read successfully.")
```

```
finally:
```

```
    print("Operation completed.")
```

What will happen if the file "data.txt" does not exist in the specified directory?

### Options:

1. The program will print :

*Error : You do not have permission to read the file.*

*Operation completed.*

2. The program will raise a *FileNotFoundError* and crash.

3. The program will print:

*Error: The file does not exist.*

*Operation completed.*

4. The program will print:

*File read successfully.*

*Operation completed.*

**Correct Answer: 3**

**Explanation:**

When the specified file does not exist, the program raises a `FileNotFoundError`. The exception is caught by the first `except` block, printing an error message. The `finally` block executes after any exceptions, ensuring "Operation completed." is printed.

**Question 4**

A school management system calculates student grades based on their marks:

- Marks  $\geq 90$ : Grade A
- Marks  $\geq 75$  and  $< 90$ : Grade B
- Marks  $\geq 50$  and  $< 75$ : Grade C
- Marks  $< 50$ : Grade F

**Question: What will be the output for marks = 68?**

```
if marks >= 90:
    print("Grade A")
elif marks >= 75:
    print("Grade B")
elif marks >= 50:
    print("Grade C")
else:
    print("Grade F")
```

**Options:**

1. Grade A
2. Grade B
3. Grade C
4. Grade F

**Answer: c) Grade C**

**Explanation:**

As our input is 68, it checks the if condition at each instance,

marks  $\geq 90$  - Not True

marks  $\geq 75$  - Not True

marks  $\geq 50$  - True ( Hence will print the grade as " Grade C")

as one of the else if condition is executed others will be ignored.

### Question 5

A programmer wants to write a program which processes numbers from 1 to 20 but skips multiples of 4 and stops if it encounters 15.

**Question: What will the following code output?**

```
for i in range(1, 21):
```

```
    if i % 4 == 0:
```

```
        continue
```

```
    if i == 15: break
```

```
    print(i, end=" ")
```

**Options:**

1. 1 2 3 5 6 7 9 10 11 13 14
2. 1 2 3 5 6 7 9 10 11 13 14 15
3. 1 2 3 5 6 7 9 10 11 13 14 16 17 18 19 20
4. 1 2 3 5 6 7 9 10 11 13

Answer: 1. 1 2 3 5 6 7 9 10 11 13 14

### Explanation

- i = 1: Not a multiple of 4, not 15 → prints 1.
- i = 2: Not a multiple of 4, not 15 → prints 2.
- i = 3: Not a multiple of 4, not 15 → prints 3.
- i = 4: Multiple of 4 → skipped (no print).
- i = 5: Not a multiple of 4, not 15 → prints 5.
- i = 6: Not a multiple of 4, not 15 → prints 6.
- i = 7: Not a multiple of 4, not 15 → prints 7.
- i = 8: Multiple of 4 → skipped (no print).
- i = 9: Not a multiple of 4, not 15 → prints 9.
- i = 10: Not a multiple of 4, not 15 → prints 10.
- i = 11: Not a multiple of 4, not 15 → prints 11.
- i = 12: Multiple of 4 → skipped (no print).
- i = 13: Not a multiple of 4, not 15 → prints 13.
- i = 14: Not a multiple of 4, not 15 → prints 14.

- `i = 15`: Equals 15 → loop breaks (stops), no printing of 15 or anything after.

The numbers printed are: 1 2 3 5 6 7 9 10 11 13 14.

- Multiples of 4 (4, 8, 12, 16, 20) are skipped.
- The loop stops at 15, so 15 and beyond (16, 17, 18, 19, 20) aren't printed.

## Question 6

### Scenario:

A travel agency maintains records of customer bookings. Each booking is stored as a tuple containing the customer's name and destination. These tuples are stored in a list. The agency wants to convert this list into a dictionary where the **customer name** is the key and the **destination** is the value. However, some customers have booked multiple destinations.

### Code Snippet:

```
bookings = [
    ("Alice", "Paris"),
    ("Bob", "New York"),
    ("Alice", "Rome"),
    ("Charlie", "Berlin"),
    ("Bob", "Tokyo")
]

# Convert to dictionary where each key has a list of destinations
travel_dict = {}

for name, destination in bookings:
    if name in travel_dict:
        travel_dict[name].append(destination)
    else:
        travel_dict[name] = [destination]

print(travel_dict)
```

What will be the output of `print(travel_dict)`?

### Options:

- A) {'Alice': ['Paris', 'Rome'], 'Bob': ['New York', 'Tokyo'], 'Charlie': ['Berlin']}
- B) {'Alice': 'Rome', 'Bob': 'Tokyo', 'Charlie': 'Berlin'}
- C) {'Alice': ['Rome'], 'Bob': ['Tokyo'], 'Charlie': ['Berlin']}
- D) {'Alice': 'Paris', 'Bob': 'New York', 'Charlie': 'Berlin'}

**Correct Answer:**

**A) {'Alice': ['Paris', 'Rome'], 'Bob': ['New York', 'Tokyo'], 'Charlie': ['Berlin']}**

**Explanation:**

- The code uses a dictionary where the value for each key is a **list**.
- If the customer already exists in the dictionary, it **appends** the new destination.
- This ensures that **all destinations booked by a customer** are stored in a list.
- The output reflects all entries grouped by customer name as expected from the logic.

### **Question 7**

An attendance system stores records in a file named attendance.txt . Each line contains a student's name and status (Present/Absent).

Question: What will the following code snippet do?

with open("attendance.txt", "a") as file:

```
    file.write("John: Present\n")
```

```
    file.write("Emily: Absent\n")
```

with open("attendance.txt", "r") as file:

```
    print(file.read())
```

**Options:**

- a) Append "John: Present" and "Emily: Absent" to attendance.txt and display the updated file contents.
- b) Overwrite the contents of attendance.txt and display the updated file contents.
- c) Display only "John: Present\nEmily: Absent" without modifying the file.
- d) Raise a FileNotFoundError.

**Answer:** a) Append "John: Present" and "Emily: Absent" to attendance.txt and display the updated file contents.

**Explanation:**

The code does two things. First, with open("attendance.txt", "a") as file: opens the file attendance.txt in **append mode** ("a"), meaning it adds new data to the end of the file without overwriting existing content. It then writes "John: Present\n" and "Emily: Absent\n" to the file. Second, with open("attendance.txt", "r") as file: opens the same file in **read**

`mode ("r")` and `print(file.read())` displays all its contents, including the newly added lines. So, the code appends those two lines to the file and then shows the updated file contents. That's why the answer is **a**).

### Question 8

A developer chooses Merge Sort for sorting a dataset, which is large and stored in external memory, such as on disk, where random access is expensive. Why might this be a good choice?

#### Options:

1. It is faster than all other sorting algorithms in all cases.
2. It has a predictable time complexity of  $O(n \log n)$ .
3. It uses less memory compared to other algorithms like Quick Sort or Heap Sort.
4. It can handle unsorted arrays better than other algorithms like Bubble Sort or Selection Sort.

#### Answer and Explanation:

- Correct option: 2
- Merge Sort guarantees consistent performance with a time complexity of  $O(n \log n)$ . However, it does require additional memory for merging subarrays.

### Question 9

A student is analyzing different sorting algorithms based on their characteristics and performance under various conditions. Which of the following statements are correct? (Select all that apply.) (MSQ)

#### Options:

1. Insertion Sort performs well for small or nearly sorted datasets.
2. Quick Sort can degrade to  $O(n^2)$  if poor pivot choices are made consistently.
3. Bubble Sort outperforms Insertion Sort for large datasets due to fewer comparisons in its best case.
4. Selection Sort always performs consistently regardless of input order but is not stable.

#### Answer and Explanation:

- Correct options: 1, 2, 4
- Insertion Sort is efficient for small or nearly sorted datasets due to its low overhead ( $O(n^2)$  worst case but close to  $O(n)$  best case). Quick Sort can degrade to quadratic complexity with poor pivot choices, while Selection Sort has consistent performance but isn't stable.

### Question 10

Aria is solving a riddle to unlock the treasure chest in an ancient temple. The lock on the chest requires her to evaluate a Python expression that combines mathematical, comparison, and logical operators.

Evaluate the following Python expression and determine the correct output:

`x = 8`

`y = 3`

`z = 10`

```
result = (x % y == 2) and (z > x + y) or (x * y < z // 2)
```

```
print(result)
```

**Options:**

A) True

B) False

C) Error

D) None

**Answer: B**

**Explanation:**

For `x = 8`, `y = 3`, `z = 10`, the expression `(x % y == 2) and (z > x + y) or (x * y < z // 2)` evaluates as:

- `x % y == 2` → `8 % 3 = 2` → True.
- `z > x + y` → `10 > 8 + 3` → `10 > 11` → False.
- `x * y < z // 2` → `8 * 3 < 10 // 2` → `24 < 5` → False.
- True and False or False → False or False → False.

So, the output is **B) False**.

### Question 11

Alex is creating a program to analyze the growth of a specific bacterial culture. The bacteria's population doubles every hour, but due to limited resources, it can't exceed 1,000,000 cells. Alex writes a recursive function to calculate the bacterial population after a given number of hours:

```
def bacterial_growth(hours, initial_population=1):
```

```
    if hours == 0:
```



```

    return initial_population

if initial_population >= 1000000:
    return 1000000

return bacterial_growth(hours - 1, initial_population * 2)

```

Alex wants to know how many hours it will take for the bacterial population to reach or exceed 1,000,000 cells, starting from a single cell.

**Question** How many hours will it take for the bacterial population to reach or exceed 1,000,000 cells? Use Alex's function in a loop to determine this value.

**(NAT)**

**Correct Answer: 20**

**Explanation** The *bacterial\_growth* function models population growth where bacteria double every hour but are capped at 1,000,000 cells. To find how long it takes to reach or exceed this value:

1. The function starts with 1 cell at hour 0.
2. Each recursive call doubles the population until it reaches or exceeds 1,000,000 cells.

| Hour | Population    |
|------|---------------|
| 1    | 2 cells       |
| 2    | 4 cells       |
| 3    | 8 cells       |
| .... | ....          |
| 19   | 524288 cells  |
| 20   | 1058576 cells |

At hour 20, the population exceeds 1,000,000 cells for the first time.

Thus, it takes 20 hours for the bacterial population to reach or exceed 1,000,000 cells, starting from a single cell.

**Question 12**

A teacher has stored the marks of all students in a list L with values from index 0 to n. Accidentally, a student modifies the code by adding the line  $L = L * 2$ , which changes the output of the list. The teacher, running out of time, doesn't fix the existing code but adds a

single line just before the `print(L)` statement to correct the output and get the original list back. Which of the following lines did the teacher add just before `print(L)`?

**Code Snippet:**

```
L = [10, 20, 30, 40] # Original list
L = L * 2           # Student's modification
# Teacher adds a line here
print(L)
```

**Options:**

- A) `L = L[:len(L)//2]`
- B) `L = L[::2]`
- C) `L = L[len(L)//2:]`
- D) `L = L[1::2]`

**Answer: A) `L = L[:len(L)//2]` C) `L = L[len(L)//2:]`**

**Explanation:**

- Initially, `L = [10, 20, 30, 40]` (let's assume this is the original list for clarity).
- The student's line `L = L * 2` duplicates the list by repeating it:
  - `L` becomes `[10, 20, 30, 40, 10, 20, 30, 40]` (length doubles from 4 to 8).
- The teacher wants the original list `[10, 20, 30, 40]` back without altering the student's line.

**1. A) `L = L[:len(L)//2]`**

- `len(L)` is 8 (after `L * 2`).
- `len(L) // 2 = 8 // 2 = 4`.
- `L[:4]` takes the first 4 elements: `[10, 20, 30, 40]`.
- **Result:** Restores the original list.
- **Correct Answer.**

**2. B) `L = L[::2]`**

- This takes every second element starting from index 0: `[10, 30, 10, 30]`.
- **Result:** Gives `[10, 30, 10, 30]`, not the original list.
- **Incorrect.**

**3. C) `L = L[len(L)//2:]`**

- `len(L) // 2 = 4`.

- `L[4:]` takes elements from index 4 to the end: `[10, 20, 30, 40]`.
- **Result:** Also restores the original list, but relies on the list being a simple repetition. If the original list had different patterns, it might fail.

Hence, Correct.

4. **D) `L = L[1::2]`**

- This takes every second element starting from index 1: `[20, 40, 20, 40]`.
- **Result:** Gives `[20, 40, 20, 40]`, not the original list.
- **Incorrect.**

### Question 13

In a small game, a player is asked to enter their score as a number. The program takes this input, doubles it, and adds 5 to calculate their final score. The player enters the number 8. What will the program output as the final score?

**Code:**

```
score = int(input("Enter your score: "))
final_score = score * 2 + 5
print(final_score)
```

**Answer : 21**

### Explanation for the Correct Answer:

Let's break it down simply:

1. **Input:**

- The player enters 8.
- `input("Enter your score: ")` takes the user's input as a string ("8").
- `int()` converts it to an integer: `score = 8`.

2. **Calculation:**

- The formula is `final_score = score * 2 + 5`.
- Substitute `score = 8`:
  - `score * 2 = 8 * 2 = 16`.
  - `16 + 5 = 21`.
- So, `final_score = 21`.

### 3. Output:

- `print(final_score)` outputs 21.

#### Why 21 is Correct:

- The program doubles the input ( $8 * 2 = 16$ ) and then adds 5 ( $16 + 5 = 21$ ).
- There are no tricks or errors here—just straightforward math based on the input 8.
- If you forget to double first and add 5 to 8 directly (like  $8 + 5 = 13$ , then  $13 * 2$ ), you'd get 26, which is wrong. The order matters: multiply first, then add.

So, when the player enters 8, the final score output by the program is **21**.

#### Question 14

A student is searching for a number in a sorted list of 100 elements. Which search algorithm would generally perform faster, and why?

#### Options:

- A) Binary Search, because it divides the search space in half each step.
- B) Linear Search, because it checks each element one by one.
- C) Binary Search, because it checks every element faster.
- D) Linear Search, because it works better on sorted lists.

#### Correct Answer:

- A) Binary Search, because it divides the search space in half each step.

#### Explanation :

##### Binary Search

- Works on a **sorted list**.
- Divides the list in half with each step (e.g., checks the middle element, then narrows to the left or right half).
- Time complexity:  $O(\log n)$ . For 100 elements, it takes at most  $\sim 7$  steps (since  $2^6 = 64$  and  $2^7 = 128$ ).

##### Linear Search

- Works on any list (sorted or unsorted).
- Checks each element one by one from start to end.
- Time complexity:  $O(n)$ . For 100 elements, it could take up to 100 steps in the worst case.

#### Why A is Correct

- Binary Search is faster for sorted lists because it reduces the search space exponentially ( $\log n$ ) compared to Linear Search's linear progression ( $n$ ).

### Why Others are Wrong

- **B:** Linear Search is slower, not faster, as it doesn't use the sorted nature of the list.
- **C:** Binary Search doesn't check every element; it skips large portions, making this reasoning incorrect.
- **D:** Linear Search doesn't benefit from a sorted list; Binary Search does.

### Question 15

A student is writing a Python program and declares two variables:  $x = 5$  and  $y = "10"$ . The student then tries to add them with  $z = x + y$ . What will happen?

#### Options:

- A)  $z$  will be 15, because Python converts the string to an integer automatically.
- B) A `TypeError` will occur, because you cannot add an integer and a string directly.
- C)  $z$  will be "510", because Python concatenates the integer and string.
- D)  $z$  will be 15, because Python treats all numbers the same regardless of type.

#### Correct Answer:

- B) A `TypeError` will occur, because you cannot add an integer and a string directly.

#### Explanation:

Here's a short explanation:

- $x = 5$  is an integer,  $y = "10"$  is a string.
- $z = x + y$  tries to add them, but Python can't mix an `int` and `str` directly.
- It throws a **`TypeError`**, not addition or concatenation.

**Why B:** `TypeError` happens because types don't match.

- **A:** No auto-conversion to 15.
- **C:** No concatenation to "510" (needs both as strings).
- **D:** Python cares about types, so not 15.

### Question 16

A data analyst is working on a project to visualize the relationships between different features in a dataset. The dataset includes variables such as age, income, and spending habits. In Seaborn, which function is specifically designed to visualize the relationship between multiple variables by creating a grid of scatter plots?

#### Options:

- A) jointplot
- B) Grid
- C) pairplot
- D) heatmap

**Correct Answer:** C) pairplot

**Explanation:** The pairplot function creates a grid of scatter plots to visualize relationships between all pairs of features in a dataset.

### Question 17

A researcher is comparing different visualization tools to understand the distribution of exam scores. When creating a violin plot in Seaborn, which aspect does it primarily visualize compared to a box plot?

**Options:**

- A) Only the mean of the data distribution
- B) The full distribution shape along with its probability density function
- C) The median value only without any distribution information
- D) Only outliers present in the dataset

**Correct Answer:** B) The full distribution shape along with its probability density function

**Explanation:** A violin plot visualizes the full distribution shape along with its probability density function, providing more insight into data distribution compared to a box plot.

### Question 18

Sarah manages a bookstore and uses NumPy to track the stock of books in different genres. She wants to check the combined stock of Science and Romance books and ensure all genres have at least 5 books.

Which NumPy concepts can be applied to solve the tasks of checking the combined stock of Science and Romance books and verifying if all genres have at least 5 books?

**Options:**

- A) Indexing and .all() method
- B) Slicing and .sum() method
- C) Indexing and .any() method
- D) Indexing, Slicing, and .all() method

**Correct Answer:** D) Indexing, Slicing, and .all() method

**Explanation:**

- Task 1 requires Indexing to select specific genres: `inventory[2] + inventory[4]`.
- Task 2 requires checking if all elements meet a condition using `.all()`: `(inventory >= 5).all()`.

### Question 19

Alice is developing a software project using object-oriented programming principles. She needs to evaluate the following statements about OOP in Python. How many of the following statements about Object-Oriented Programming (OOP) in Python are true?

#### Options:

- A) In Python, a class can only inherit from one class at a time.
- B) Python allows multiple inheritance, where a class can inherit from more than one class.
- C) An instance method in a class must always take self as the first parameter.
- D) A class in Python can have multiple constructors.
- E) In Python, you can call a method of a superclass using super().

**Correct Answer:** 3

**Explanation:** Statements B, C, and E are true.

### Question 20

Emma is analyzing customer feedback data stored in a Pandas DataFrame named `df_feedback`. She suspects there are missing values and wants to calculate the total number of NaN values.

Which line of code will help Emma find the total number of NaN values in the DataFrame `df_feedback`?

#### Options:

- A) `df_feedback.isnull().sum(axis=1).sum()`
- B) `df_feedback.isnull().sum().sum()`
- C) `df_feedback.isna().any(axis=1).sum()`
- D) `df_feedback.isna().sum(axis=1).sum()`

**Correct Answer**

B: `df_feedback.isnull().sum().sum()`

#### Explanation:

1. `df_feedback.isnull()` generates a DataFrame with True for NaN values.
2. `.sum()` applied once counts NaNs per column.
3. `.sum()` applied again gives the total count across all columns.

### Question 21

Sophia is working on a dataset named `df_employees`, which contains information about employees in a company. She wants to analyze the dataset for employees with a salary greater than 75,000 and count how many employees belong to the "IT" department and joined after 2015. Which code snippet will help Sophia identify employees with a salary

greater than 75,000 and count how many employees belong to the "IT" department and joined after 2015?

**Options:**

A)

# Task 1

```
high_salary = df_employees[df_employees['Salary'] > 75000]
```

# Task 2

```
it_after_2015 = df_employees[(df_employees['Department'] == 'IT') &  
(df_employees['Joining Year'] > 2015)]
```

```
count = it_after_2015.shape[0]
```

B.

# Task 1

```
high_salary = df_employees['Salary'] > 75000
```

# Task 2

```
count = df_employees['Department'].value_counts()['IT'] & (df_employees['Joining Year'] >  
2015).sum()
```

C.

# Task 1

```
high_salary = len(df_employees[df_employees['Salary'] > 75000])
```

# Task 2

```
count = (df_employees['Department'] == 'IT') & (df_employees['Joining Year'] > 2015).sum()
```

D.

# Task 1

```
high_salary = df_employees[df_employees['Salary'] > 75000].shape[0]
```

# Task 2

```
count = df_employees.query("Department == 'IT' and `Joining Year` > 2015").shape[0]
```

**Correct Answer: A**

**Explanation:**

- Task 1 filters rows where Salary is greater than 75,000 using a condition: `df_employees['Salary'] > 75000`.
- Task 2 uses a combined condition with `&` to filter rows where Department is "IT" and Joining Year is greater than 2015. Use `.shape[0]` to count these rows.



### Question 22

A quality control manager is checking the probability of drawing a red ball followed by a blue ball from a box containing 5 red balls, 3 blue balls, and 2 green balls. What is the probability that the first ball drawn is red and the second ball is blue when two balls are drawn without replacement from a box containing 5 red balls, 3 blue balls, and 2 green balls?

#### Options:

- A)  $3/20$
- B)  $15/90$
- C)  $5/18$
- D)  $1/6$

**Correct Answer:** D)  $1/6$

#### Explanation:

- Probability of drawing a red ball first is  $5/10 = \frac{1}{2}$ .
- Probability of drawing a blue ball next is  $3/9 = \frac{1}{3}$ .
- Combined probability is  $\frac{1}{2} * \frac{1}{3} = 1/6$ .

### Question 23

A software developer is evaluating object-oriented programming principles for a project. Which of the following statements about classes and objects in Python are correct?

#### Options:

- A) A class is a blueprint for creating objects, while an object is an instance of a class.
- B) Python does not allow a class to have class variables, only instance variables.
- C) The `str` method is used to provide a string representation of an object.

**Correct Answer:** A,C

**Explanation:** Statements A, C are correct.

### Question 24

A data analyst is studying the effect of doubling each observation in a dataset on its mean, median, mode, and variance. What happens to the mean, median, mode, and variance of a dataset when each observation is doubled?

#### Options:

- A) Mean doubles, Median doubles, Mode doubles, Variance doubles
- B) Mean doubles, Median remains the same, Mode doubles, Variance quadruples
- C) Mean doubles, Median doubles, Mode doubles, Variance remains the same
- D) Mean doubles, Median doubles, Mode doubles, Variance increases by a factor of 4

**Correct Answer:** D) Mean doubles, Median doubles, Mode doubles, Variance increases by a factor of 4

**Explanation:**

- **Mean:** When each observation is doubled, the sum of all observations also doubles, while the number of observations remains the same. Thus, the mean doubles.
- **Median:** The median is the middle value when data is ordered. Doubling each observation will double the median.
- **Mode:** The mode is the most frequently occurring value. Doubling each observation will double the mode.
- **Variance:** Variance measures the spread of data from its mean. When each observation is scaled by a constant (in this case, doubled), the variance is scaled by the square of that constant. Therefore, the variance increases by a factor of  $2^2=4$ .

#### Question 25

A quality control engineer is using statistical methods to monitor product weights.

What is the purpose of marking critical weight boundaries on a bell curve in quality control?

**Options:**

- A) To visualize the distribution of sample weights
- B) To determine the sample size
- C) To set the threshold for rejecting the null hypothesis
- D) To calculate the standard error

**Correct Answer:** C) To set the threshold for rejecting the null hypothesis

**Explanation:** Marking critical weight boundaries helps visualize the threshold beyond which the null hypothesis is rejected.

#### Question 26

A retailer is checking if the average weight of products received differs from the product weight claimed by the wholesaler.

What statistical test is being performed when checking if the average weight of products received by a retailer differs from the product weight claimed by the wholesaler?

**Options:**

- A) Comparing two sample means
- B) Testing if a sample mean differs from a known population mean
- C) Determining the sample size
- D) Calculating the standard error

**Correct Answer:** B) Testing if a sample mean differs from a known population mean

**Explanation:** This scenario involves comparing a sample mean (the average weight of products received) to a known population mean (the product weight claimed by the wholesaler). This is typically done using a z-test if the population variance is known or a t-test if the population variance is unknown. The goal is to determine if there is a statistically significant difference between the sample mean and the known population mean.

### Question 27

Your classmate is designing a seating chart for a theater and needs your help. The chart is represented as a list of lists:

```
seats = [["A1", "A2", "A3"], ["B1", "B2", "B3"], ["C1", "C2", "C3"]]
```

Each sublist represents a row of seats. She wants to rotate the chart so columns become rows — essentially finding the transpose.

What will the seating chart look like after the transformation?

#### Options:

- A) [["A1", "B1", "C1"], ["A2", "B2", "C2"], ["A3", "B3", "C3"]]
- B) [["C1", "B1", "A1"], ["C2", "B2", "A2"], ["C3", "B3", "A3"]]
- C) [["A3", "B3", "C3"], ["A2", "B2", "C2"], ["A1", "B1", "C1"]]
- D) [["A1", "A2", "A3"], ["B1", "B2", "B3"], ["C1", "C2", "C3"]]

**Correct Answer:** A)

#### Explanation:

- Transposing means swapping rows with columns.
- Original rows: ["A1", "A2", "A3"], ["B1", "B2", "B3"], ["C1", "C2", "C3"]
- Transposed: ["A1", "B1", "C1"], ["A2", "B2", "C2"], ["A3", "B3", "C3"]

### Question 28

Your teammate is working on a physics simulation project. She stores a 3D position vector of a drone as:

```
position = [12, -5, 9]
```

She needs to reverse the direction of the vector for the simulation. Which of the following correctly represents the reversed direction vector?

#### Options:

- A) [-12, 5, -9]
- B) [12, 5, 9]
- C) [-12, -5, -9]
- D) [9, -5, 12]

**Correct Answer:** A) [-12, 5, -9]

**Explanation:**

- Reversing the direction of a vector means multiplying each component by -1.
- Original vector: [12, -5, 9]
- Correct reversed vector: [-12, 5, -9]

### Question 29

Sarah is studying a simple Markov Chain model for customer behavior in an online store. The model has only three states: "Browsing", "Adding to Cart", and "Checkout." The transition probabilities are:

From "Browsing": 70% stay browsing, 20% move to "Adding to Cart" 10% go to "Checkout".

From "Adding to Cart": 30% return to browsing, 50% stay adding items, x% proceed to checkout.

From "Checkout": 40% return to browsing, 10% go back to adding items, 50% complete the purchase and exit.

What is the value of x?

**Options:**

- A) 0.30
- B) 0.20
- C) 0.50
- D) 0.10

**Correct Answer:** B) 0.20

**Explanation:** The transition probability from "Adding to Cart" to "Checkout" is  $1 - P(\text{"Adding to Cart" | "Adding to Cart"}) - P(\text{"Browsing" | "Adding to Cart"}) = 1 - 30\% - 50\% = 20\%$  or 0.20.

### Question 30

Your friend is given an assignment, and she comes to you for help. Below is the question she wants answered.

Given two matrices:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

What is the transpose of matrix B?

**Options:**

- A)  $\begin{bmatrix} 17 & 23 \\ 39 & 53 \end{bmatrix}$
- B)  $\begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$
- C)  $\begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$  D)  $\begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix}$

**Correct Answer:** D)

**Explanation:**

- Transpose of a matrix means swapping its rows and columns.
- So for  $B = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$ , the transpose  $B^T$  is:  $B^T = \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix}$