Editorial-Mock : Mid Module A Quiz - 1

Question 1.

You are programming a drone delivery system to locate items in different storage zones of a warehouse.

- **Zone A**: Items are randomly scattered (unsorted).
- **Zone B**: Items are sorted by weight in ascending order.

The drone's battery consumption depends on the number of comparisons it makes during a search. The drone needs to find an item with **ID P567**:

- Zone A has 500 items.
- Zone B has 1024 items.
- Each comparison in **Zone A** costs 2 units of battery, and each comparison in **Zone B** costs 1 unit of battery.

What is the total battery consumption in the **worst case** if the drone uses linear search in Zone A and binary search in Zone B?

Correct Answer: 1010 units

Question 2.

You are working with a very large dataset where almost all elements are identical, except for a few outliers. You need to choose the best sorting algorithm to handle this case efficiently.

Which sorting algorithm is most likely to perform the best in this scenario?

A. Bubble Sort B. Selection Sort C. Quick Sort D. Insertion Sort

Correct Answer: C. Quick Sort

Question 3: Python Grading Logic

You are creating a grading system for a school. The grading rules are as follows:

- If the score is **90 or above**, the grade is **"A"**.
- If the score is **80 to 89** (inclusive of 80, but less than 90), the grade is **"B"**.
- If the score is **70 to 79** (inclusive of 70, but less than 80), the grade is **"C"**.
- If the score is **60 to 69** (inclusive of 60, but less than 70), the grade is **"D"**.
- If the score is **below 60**, the grade is **"F"**.

Which of the following Python code snippets correctly implement the grading logic? (Assume score is a positive integer.)

Option (a)

```
if score <= 60:
    grade = "F"
elif score <= 70:
    grade = "D"
elif score <= 80:
    grade = "C"
elif score <= 90:
    grade = "B"
else:
    grade = "A"
```

Option (b)

```
if score < 60:
    grade = "F"
elif score < 70:
    grade = "D"
elif score < 80:
    grade = "C"
elif score < 90:
    grade = "B"
else:
    grade = "A"
```

Option (c)

```
if score > 90:
    grade = "A"
elif score >= 80 and score < 90:
    grade = "B"
elif score >= 70 and score < 80:
    grade = "C"
elif score >= 60 and score < 70:
    grade = "D"
else:
    grade = "F"
```

Option (d)

```
if score >= 90:
    grade = "A"
elif score >= 80:
    grade = "B"
elif score >= 70:
    grade = "C"
elif score >= 60:
    grade = "D"
else:
    grade = "F"
```

Answer: B, D

Question 4.

You are creating a program that counts how many even and odd numbers are present in a list of integers. But, there's a twist, if a negative number is encountered while iterating through the loop, then the loop should terminate early and only print counts of even and odd numbers till then.

Question: Which of the following code snippets correctly implements this counting logic?

Option (a)

```
numbers = [2, -4, -7, -5, -8]
even_count = odd_count = 0

for num in numbers:
    if num < 0:
        continue
    if num % 2 == 0:
        even_count += 1
    else:
        odd_count += 1

print("Even numbers:", even_count)
print("Odd numbers:", odd_count)
```

Option (b)

```
numbers = [2, 4, 7, -1, 5, 8]
even_count = 0
odd_count = 0

for num in numbers:
    if num < 0:
        break
    if num % 2 == 0:
        even_count += 1
    else:
        odd_count += 1

print("Even numbers:", even_count)
print("Odd numbers:", odd_count)
```

Option (c)

```
numbers = [2, -4, -7, -5]
even_count = odd_count = 0

while True:
    for num in numbers:
        if num < 0:
            break
        if num % 2 == 0:
            even_count += 1
        else:
            odd_count += 1

print("Even numbers:", even_count)
print("Odd numbers:", odd_count)
```

Answer: B

Question 5.

You are developing a simple number guessing game. The program will ask the user to guess a number between 1 and 10. The user has up to 3 attempts to guess the correct number. If the user guesses correctly, the game ends, but if they exceed 3 attempts, a message is displayed stating that they have lost.
Question:
Which of the following code snippets correctly implements this number guessing logic?

Option (a)

```
import random

number_to_guess = random.randint(1, 10)
attempts = 0

while attempts < 3:
    guess = int(input("Guess a number between 1 and 10: "))
    if guess == number_to_guess:
        print("Congratulations! You've guessed the number.")
        break
    attempts += 1
else:
    print("Sorry, you've lost! The number was", number_to_guess)
```

Option (b)

```
import random
```

```
number_to_guess = random.randint(1, 10)

for attempts in range(3):
    guess = int(input("Guess a number between 1 and 10: "))
    if guess == number_to_guess:
        print("Congratulations! You've guessed the number.")
        break
else:
    print("Sorry, you've lost! The number was", number_to_guess)
```

Option (c)

```
import random

number_to_guess = random.randint(1, 10)
attempts = 0

while True:
    guess = int(input("Guess a number between 1 and 10: "))
    if guess == number_to_guess:
        print("Congratulations! You've guessed the number.")
        break
    attempts += 1
    if attempts >= 3:
        print("Sorry, you've lost! The number was", number_to_guess)
        break
```

Option (d)

```
import random
```

```
number_to_guess = random.randint(1, 10)
print(number_to_guess)
for attempts in range(4):
    guess = int(input("Guess a number between 1 and 10: "))
    if guess == number_to_guess:
        print("Congratulations! You've guessed the number.")
        break
print("Sorry, you've lost! The number was", number_to_guess)
```

Answer: A, B, C

Question 6.

You are developing a simple movie rating application that restricts access based on age. The rules are as follows:
If the user is under 13 years old, they can only watch movies rated G (General Audience).
If the user is between 13 (included) and 18 (excluded) years old, they can watch movies rated G and PG (Parental Guidance).
If the user is 18 years or older, they can watch movies rated G, PG, PG-13 (Parents Strongly Cautioned), and R (Restricted).
Question:
Which of the following Python code snippets correctly implements this age-based movie rating logic? (**Given**, age is a positive real number)
(MSQ)

Option (a)

```
if age < 13:
    rating_access = ["G"]
elif age >= 13 and age < 18:
    rating_access = ["G", "PG"]
else:
    rating_access = ["G", "PG", "PG-13", "R"]
```

Option (b)

```
if age <= 13:
    rating_access = "G only"
```

```
elif age >= 13 and age < 18:
    rating_access = "G and PG only"
else:
    rating_access = "G, PG, PG-13, and R"
```

Option (c)

```
if age < 13:
    rating_access = "G only"
elif age < 18:
    rating_access = "G and PG only"
else:
    rating_access = "G, PG, PG-13, and R"
```

Option (d)

```
if age > 18:
    rating_access = ["G", "PG", "PG-13", "R"]
elif age >= 13 and age < 18:
    rating_access = ["G", "PG"]
else:
    rating_access = ["G"]
```

Answer: A, C

Question 7.

You want to write some text to a new file called example.txt. Which of the following code snippets correctly creates the file and writes the text "Hello, World!" to it?

Option (a)

```
with open('example.txt', 'r') as file:
    file.write('Hello, World!')
```

Option (b)

```
file = open('example.txt', 'w')
file.write('Hello, World!')
file.close()
```

Option (c)

```
file = open('example.txt', 'a')
file.write('Hello, World!')
file.close()
```

Option (d)

```
with open('example.txt', 'w') as file:
   file.write('Hello, World!')
```

Answer: B,C,D

Question 8 and 9

Chef Ravi is creating a recipe app. In one feature, the app reads an ingredient list from a file. If the file is missing, the program should handle the error. In another feature, Ravi is adding a calculator to divide quantities of ingredients. However, he realizes the program crashes when someone tries to divide by zero. Ravi needs to handle both these scenarios gracefully.

Question 8:

Which exceptions should Ravi handle in the following cases?

1. The file containing the ingredients is missing.
2. The program attempts to divide by zero.

Options:

- A. FileNotFoundError
- B. TypeError
- C. ZeroDivisionError
- D. ValueError

Correct Answer: A, C

Question 9:

Chef Ravi uses the following code to calculate ingredient ratios:

```
def divide_ingredients(a, b):
        try:
                result = a / b
        except ZeroDivisionError:
            result = 0
        return result
```

Now, Ravi extends the function to handle cases where either of the inputs is not a number (e.g., a string or None). If such an invalid input is provided, the function should return -1.

The function is called with a = 45 and b = '0'. What will the function return?

Correct Answer: -1

Question 10 and 11

Sophia is building a payroll system that processes employee data from an input file. Each entry is expected to be a dictionary with the keys name, age, and salary. If any required data is missing or if the salary is non-numeric, the system should handle these cases gracefully. Additionally, Tom is building a secure file uploader that only accepts .txt, .csv, and .json files. If the uploaded file is of an invalid type or there are permission issues, the system must raise an appropriate error. Both systems need to handle these exceptions to prevent crashes.

Question 10 (Multi-Correct):

Which exceptions should be handled in the following scenarios? Select all that apply:

1. A dictionary entry is missing a required key (name, age, or salary).
2. The salary field contains a non-numeric value.
3. The file uploaded has an invalid extension (not .txt, .csv, or .json).

Options:

- A. KeyError
- B. TypeError
- C. ValueError
- D. InvalidFileTypeError

Correct Answer: A, C, D

Question 11 (Single Correct):

What exception should be raised if the file uploaded is of an invalid type (i.e., not .txt, .csv, or .json)?

Options:

- A. InvalidFileTypeError
- B. FileNotFoundError

- C. PermissionError
- D. ValueError

Correct Answer: A

Question 12: Watering Crops

A farmer needs to measure the time it takes to water all the crops in a straight line. Each crop takes exactly 1 second to water, and there are n crops.

What is the time complexity of the watering process?

- A) O(1)
- B) O(n)
- C) O(n^2)
- D) O(log n)

Answer: B

Question 13

A bakery keeps track of the number of cupcakes sold each day. The owner calculates the total cupcakes sold this week using the following Python code:

```
monday = 45

tuesday = 30

wednesday = 50

thursday = 40

friday = 35

saturday = 60

sunday = 55


total_cupcakes = monday + tuesday + wednesday + thursday + friday + saturday + sunday

print(total_cupcakes)
```

What will the output be if the code is executed?

- A. 305
- B. 315
- C. 275

- D. 285

Answer: B

Q14: Updating the Team List

Your team currently consists of the following members:

team = ["Alice", "Bob", "Charlie", "Diana"]

A new member, "Eve", has joined the team. Your task is to add her to the list. Which of the following Python statements will achieve this?

- A) team.append("Eve")
- B) team.add("Eve")
- C) team.insert("Eve")
- D) team.push("Eve")

Answer: A

Q15: Fixed Schedule Planning

Your manager has fixed a meeting schedule for the week as follows:

schedule = ("Monday", "Wednesday", "Friday")

You want to check if "Friday" is included in the schedule. Which of the following statements is correct?

- A) "Friday" in schedule
- B) schedule.contains("Friday")
- C) schedule.index("Friday")
- D) schedule.find("Friday")

Answer: A

Question 16: Employee Directory

The company maintains a simple directory for each employee as shown below:

```
employee = {
   "name": "Alice",
   "position": "Software Engineer",
   "skills": ["Python", "SQL", "Cloud Computing"]
}
```

You need to add a new skill, "Machine Learning", to Alice's skills. Which of the following statements will achieve this?

- A) employee.skills.append("Machine Learning")
- B) employee["skills"].append("Machine Learning")
- C) employee["skills"] = "Machine Learning"
- D) employee.skills.add("Machine Learning")

Answer: B

Question 17

A library fines users for late book returns. The fine is calculated at $2 per day. If the fine exceeds $20, the librarian caps it at $20. The code snippet is as follows:

```
days_late = 12
fine_per_day = 2

total_fine = days_late * fine_per_day
capped_fine = total_fine if total_fine <= 20 else 20

print(capped_fine)
```

What will be the output of the code?

- A. 12
- B. 24
- C. 20
- D. 18

Answer: C

Question 18 (Medium MCQ)

A teacher wants to keep a record of students who scored well in a test. She writes a Python program to read a file containing student names and their scores, and create a new file with the program given as follows:

```
# Reading data from the input file
```

```
with open("scores.txt", "r") as input_file:
    lines = input_file.readlines()

with open("above_80.txt", "w") as output_file:
    for line in lines:
        name, score = line.strip().split(",")
        if int(score) > 80:
            output_file.write(name + "\n")
```

Content of scores.txt:

```
Alice,78
Bob,85
Charlie,92
David,76
Eve,88
```

After running the program, what will be the content of the file above_80.txt?

A.

```
Bob
Charlie
Eve
```

B.

```
Alice
Bob
Charlie
Eve
```

C.

```
Charlie
Eve
Bob
```

B.

```
Bob
Charlie
```

David
Eve

Answer: A

Question 19: Loop with Conditional

A runner records her speed at n checkpoints during a race. She wants to identify the checkpoints where her speed was greater than the average speed:

```
speeds = [5, 10, 7, 8, 9] # Assume n checkpoints

average_speed = sum(speeds) / len(speeds)

for speed in speeds:

  if speed > average_speed:

    print(speed)
```

Time and Space Complexity Analysis:

- A) Time: O(n), Space: O(1)
- B) Time: O(n), Space: O(n)
- C) Time: O(n^2), Space: O(n)
- D) Time: O(log n), Space: O(log n)

Answer: A

Question 20 (Function) - Moderate - Multi Correct

In a quiet village, there is a wise old wizard known as the "Loop Sorcerer." He challenges travelers to calculate the factorial of a number n using a loop. However, the task has a catch! Your task is to figure out which of the following statements are correct based on the behavior of the two functions, given these special conditions.

```
def special_factorial_1(n):

  if n < 0:

    return "Error: Negative numbers do not have factorials"

  fact = 1

  for i in range(1, n+1):

    fact *= i

    if i == 5:

      break
    return fact
def special_factorial_2(n):
    if n < 0:
        return "Error: Negative numbers do not have factorials"
    fact = 1
    for i in range(1, n+1):
        if i%5 != 0:
            fact*=i
    return fact
```

Options:

- A) The number of positive integers less than 10 that give the same output for these two functions is 4.
- B) special_factorial_2(10)*10*5 = 10!
- C) special_factorial_1(10)*10*9*8*7*6 = 10!
- D) special_factorial_2(6)/special_factorial_1(4) = 6.0

Correct Answer: A, B, C, D

Question 21: The Enchanted Puzzle of Truths 🔮

In the mystical land of PyLand, the magic of **truth tables** is used to determine the outcomes of various spells. The kingdom's most powerful wizard has created a complex puzzle using logical operators. The wizard defined the following magical variables:

```
A = True
B = False
C = True
D = False
```

Using these variables, the wizard wants you to determine the outcome of several magical expression:

```
print(not(A and (B or C)), not(D or (A and C)), (A or B) and (C and not(D)), not(C) or (A and D))
```

What will be the output of the given code?

Options:

A) **True True True False**
B) **False True False True**
C) **True False False True**
D) **False False True False**

Correct Answer: Option D

Question 22: 🧙 Wizard's Potion Brewing 🧪

A wizard is brewing potions in his magical lab, and he has a specific process for checking ingredients. Each ingredient has a potency level, and based on this level, the wizard performs different actions:

1. If the potency level is -1, the wizard encounters a *spoiled* ingredient and skips to the next one.
2. If the potency level is 0, the wizard immediately **stops** brewing and leaves the lab.
3. If the potency level is greater than 100, the wizard **pauses** and carefully analyzes the ingredient, but continues to the next one.
4. If the potency level is between 1 and 5 (inclusive), the wizard **adds** the ingredient to the potion and prints: "Added ingredient with potency X".
5. If the potency level is between 6 and 15 (inclusive), the wizard **stirs** the potion and prints: "Stirring potion with ingredient X".
6. If the potency level is between 16 and 30 (inclusive), the wizard **scans** the ingredient and prints: "Scanned ingredient with potency X".
7. If the potency level is between 31 and 50 (inclusive), the wizard **adjusts** the potion and prints: "Adjusting potion with ingredient X".
8. If the potency level is between 51 and 100 (inclusive), the wizard **dilutes** the potion and prints: "Diluting potion with ingredient X".
9. The wizard keeps track of the number of spoiled ingredients (-1), and if he encounters three consecutive spoiled ingredients, he **stops brewing** and prints "Too many spoiled ingredients!".

The list of ingredients is represented by the following list of potency levels:

ingredients = [5, 8, -1, 3, 0, 120, 15, 18, 7, -1, 100, 25, -1, 50, 200, 10]

The wizard follows this code to process the ingredients:

```
spoiled_count = 0

for potency in ingredients:

  if potency == -1:
      spoiled_count += 1
      if spoiled_count == 3:
          print("Too many spoiled ingredients!")
          break
      continue
  elif potency == 0:
      break
  elif potency > 100:
      pass
  elif 1 <= potency <= 5:
      print(f"Added ingredient with potency {potency}")
  elif 6 <= potency <= 15:
      print(f"Stirring potion with ingredient {potency}")
  elif 16 <= potency <= 30:
      print(f"Scanned ingredient with potency {potency}")
  elif 31 <= potency <= 50:
      print(f"Adjusting potion with ingredient {potency}")
  elif 51 <= potency <= 100:
      print(f"Diluting potion with ingredient {potency}")
```

What will be the output of this code?

- A)
- Added ingredient with potency 5
- Stirring potion with ingredient 8
- Added ingredient with potency 3
- B)
- Added ingredient with potency 5
- Stirring potion with ingredient 8
- Added ingredient with potency 3
- Stopped brewing the potion
- C)

- Added ingredient with potency 5
- Stirring potion with ingredient 8
- Too many spoiled ingredients!
- D)
- Added ingredient with potency 5
- Stirring potion with ingredient 8
- Added ingredient with potency 3
- Stopped brewing the potion

Correct answer: A

Question 23: The Enchanted Garden of Magical Plants 🌱✨

A wizard has a magical garden with different plants. Each plant has a specific number of petals, and the wizard is using a magical algorithm to classify plants based on the number of petals and their status. The algorithm applies specific actions based on the conditions:

- If a plant has **3 petals**, the wizard **does nothing** (i.e., pass).
- If a plant has **more than 5 petals**, and it is divisible by **7**, the wizard **skips** the plant (i.e., continue).
- If a plant has **less than or equal to 3 petals**, the wizard **counts** it as a magical plant and prints its status.
- If the wizard has processed **more than 10 magical plants**, he **stops** the classification process (i.e., break).

The garden has the following plants (with petals count):

```
plants = [2, 3, 5, 7, 8, 3, 10, 14, 16, 21, 1, 3, 12, 15, 2, 4, 9, 11, 17, 23, 25]

magical_plants = 0


for plant in plants:

  if plant == 3:

    pass  # Do nothing for plants with 3 petals

  elif plant > 5 and plant % 7 == 0:

    continue  # Skip plants that are divisible by 7 and have more than 5 petals

  elif plant <= 3:
    magical_plants += 1
    print(f"Magical plant with {plant} petals!")
  if magical_plants > 10:
    break  # Stop after processing 10 magical plants
```

What will be the output of the code?

- A)
- Magical plant with 2 petals!
- Magical plant with 1 petals!
- Magical plant with 2 petals!
- B)
- Magical plant with 2 petals!
- Magical plant with 1 petals!
- Magical plant with 3 petals!
- C)
- Magical plant with 2 petals!
- Magical plant with 1 petals!
- Magical plant with 2 petals!
- Magical plant with 3 petals!
- D)
- Magical plant with 2 petals!
- Magical plant with 1 petals!
- Magical plant with 2 petals!
- Magical plant with 4 petals!

Correct answer: A

Question 24: The Wizard’s Magical Garden 🌿 ✨

A wizard has a magical garden, where he stores information about plants in a nested dictionary using **dictionary comprehension**. Each plant has a name, magical properties, and an associated element based on the name’s characters. The wizard uses the following code to calculate the magical value, rarity, and element of each plant:

```
plants = ['Moonflower', 'Dragonroot', 'Phoenix Feather', 'Mandrake', 'Elven Herb', 'Violet Petal', 'Fireblossom']
garden = {
    plant: {
        'magical_value': len(plant) if len(plant) % 2 != 0 else i * 3,
        'rarity': 'rare' if len(plant) >= 8 else 'common',
        'element': 'Earth' if 'o' in plant else ('Air' if 'a' in plant else 'Water')
    }
    for i, plant in enumerate(plants)
}
print(garden)
```

The code processes the **plants** list and generates a dictionary

What will be the output of the code?

- A)
- {
- 'Moonflower': {'magical_value': 0, 'rarity': 'rare', 'element': 'Earth'},
- 'Dragonroot': {'magical_value': 3, 'rarity': 'rare', 'element': 'Earth'},
- 'Phoenix Feather': {'magical_value': 15, 'rarity': 'rare', 'element': 'Earth'},
- 'Mandrake': {'magical_value': 9, 'rarity': 'rare', 'element': 'Air'},
- 'Elven Herb': {'magical_value': 12, 'rarity': 'rare', 'element': 'Water'},
- 'Violet Petal': {'magical_value': 15, 'rarity': 'rare', 'element': 'Earth'},
- 'Fireblossom': {'magical_value': 18, 'rarity': 'rare', 'element': 'Air'}
- }
- B)
- {
- 'Moonflower': {'magical_value': 0, 'rarity': 'rare', 'element': 'Earth'},
- 'Dragonroot': {'magical_value': 3, 'rarity': 'rare', 'element': 'Earth'},
- 'Phoenix Feather': {'magical_value': 15, 'rarity': 'rare', 'element': 'Earth'},
- 'Mandrake': {'magical_value': 9, 'rarity': 'rare', 'element': 'Air'},
- 'Elven Herb': {'magical_value': 12, 'rarity': 'rare', 'element': 'Water'},
- 'Violet Petal': {'magical_value': 15, 'rarity': 'rare', 'element': 'Earth'},

- 'Fireblossom': {'magical_value': 11, 'rarity': 'rare', 'element': 'Earth'}
- }
- C)
- {
- 'Moonflower': {'magical_value': 0, 'rarity': 'rare', 'element': 'Earth'},
- 'Dragonroot': {'magical_value': 3, 'rarity': 'rare', 'element': 'Earth'},
- 'Phoenix Feather': {'magical_value': 15, 'rarity': 'rare', 'element': 'Earth'},
- 'Mandrake': {'magical_value': 9, 'rarity': 'rare', 'element': 'Air'},
- 'Elven Herb': {'magical_value': 12, 'rarity': 'rare', 'element': 'Water'},
- 'Violet Petal': {'magical_value': 15, 'rarity': 'rare', 'element': 'Earth'},
- 'Fireblossom': {'magical_value': 12, 'rarity': 'rare', 'element': 'Earth'}
- }
- D)
- {
- 'Moonflower': {'magical_value': 0, 'rarity': 'rare', 'element': 'Earth'},
- 'Dragonroot': {'magical_value': 3, 'rarity': 'rare', 'element': 'Earth'},
- 'Phoenix Feather': {'magical_value': 15, 'rarity': 'rare', 'element': 'Air'},
- 'Mandrake': {'magical_value': 9, 'rarity': 'rare', 'element': 'Air'},
- 'Elven Herb': {'magical_value': 12, 'rarity': 'rare', 'element': 'Water'},
- 'Violet Petal': {'magical_value': 15, 'rarity': 'rare', 'element': 'Earth'},
- 'Fireblossom': {'magical_value': 18, 'rarity': 'rare', 'element': 'Air'}
- }

Correct answer: B

Question 25: 🧩 The Puzzle of the Lost Artifacts

A renowned archaeologist has recently discovered a collection of ancient artifacts. Each artifact has a name, a year of discovery, and a rarity level. The archaeologist wishes to perform various operations such as:

1. Adding artifacts to the collection.
2. Sorting the artifacts based on the year of discovery.
3. Retrieving artifacts with specific rarity.

You are tasked with assisting the archaeologist by analyzing the following code that performs these operations:

```
def manage_artifacts(collection, **filters):
    sorted_collection = sorted(collection, key=lambda x: x['year'])

    filtered_artifacts = []
    for artifact in sorted_collection:
        match = True
        for key, value in filters.items():
            if artifact[key] != value:
                match = False
                break
        if match:
            filtered_artifacts.append(artifact)

    return filtered_artifacts

# List of artifacts
artifacts = [
    {'name': 'Moonstone', 'year': 1960, 'rarity': 'rare'},
    {'name': 'Phoenix Feather', 'year': 1985, 'rarity': 'legendary'},
    {'name': 'Dragonroot', 'year': 1990, 'rarity': 'rare'},
    {'name': 'Mandrake', 'year': 1975, 'rarity': 'common'},
    {'name': 'Elven Herb', 'year': 2000, 'rarity': 'rare'},
    {'name': 'Violet Petal', 'year': 1980, 'rarity': 'legendary'}
]

# Querying the collection with filters
filtered_artifacts = manage_artifacts(artifacts, rarity='rare', year=1985)
for artifact in filtered_artifacts:
    print(f"Artifact: {artifact['name']}, Year: {artifact['year']}, Rarity: {artifact['rarity']}")
```

What will be the output when the code is executed? A)

Artifact: Phoenix Feather, Year: 1985, Rarity: legendary

B)

Artifact: Dragonroot, Year: 1990, Rarity: rare

C)

Artifact: Moonstone, Year: 1960, Rarity: rare

Artifact: Dragonroot, Year: 1990, Rarity: rare

Artifact: Elven Herb, Year: 2000, Rarity: rare

D) None of these

Correct answer: D

Question 26

A warehouse has sorted inventory IDs. To locate an item efficiently, they use binary search. What is the time complexity of finding an item in a dataset of size 1,048,576?
a) O(n)
b) O(log n)
c) O(n^2)
d) O(1)

Answer: O(log n)

Question 27

You are trying to sort data and you want to ensure that elements with equal values retain their original relative order after sorting. Which sorting algorithm should you choose?

Options:

1. Quick Sort
2. Selection Sort
3. Merge Sort
4. Heap Sort

Correct Answer: Merge Sort

Question 28

A dataset is already nearly sorted, with only a few elements out of place. Which sorting algorithm would likely perform best in this scenario?

Options:

1. Merge Sort
2. Quick Sort
3. Insertion Sort
4. Selection Sort

Correct Answer: Insertion Sort

Question 29

Arun is trying to find the result of an expression. Can you help him by finding the output of the following code?

```
result = 1 + 3 * 2 ** 2 - 1 + 10 // 5 + bool(-1)
print(result)
```

Options:

1. 13
2. Error
3. 17
4. 15

Correct Answer: 15

Question 30

Varun is learning input statement. He wrote a Python statement z = input(), ran the code, and entered -1.
Identify the datatype of z.

Options:

1. int
2. str
3. list
4. bool

Correct Answer: str