2 questions of MCQ with answers(1mark):

1. Given two variables, **j = 6** and **g = 3.3**. If both normal division and floor division operators were used to divide **j** by **g**, what would be the data type of the value obtained from the operations?

   a) int, int
   b) float, float
   c) float, int
   d) int, float

2. Let a = 5 (101 in binary) and b = 3 (011 in binary). What is the result of the following operation?

```
a=5
b=3
print(a&b)
```

   a) 3
   b) 7
   c) 5
   d) 1

2 Questions of descriptive type with answers (5marks or 7marks):

1. Consider the following Python code snippet executed in the Spyder IDE:

```
a = 15
b = 4
c = a / b
d = a // b
e = a % b
f = a ** b
print(c, d, e, f)
```

   (a) Write the output of the program.
   (b) Explain why `c` and `d` are different even though both perform division.

   Answer: (a) Output: 3.75 3 3 50625

   (b) **Explanation:**

- `a / b = 15 / 4 = 3.75` → Division (`/`) in Python **always returns float**, even with integers.
- `a // b = 15 // 4 = 3` → Floor Division truncates the decimal part, returning only the integer quotient.
- `a % b = 15 % 4 = 3` → Modulus gives the remainder.
- `a ** b = 15 ** 4 = 50625` → Exponentiation operator.

Thus, `/` and `//` behave differently in Python, making explicit control over division possible.

2. Differentiate between mutable and immutable data types in Python with examples. Why is understanding this distinction important for data science applications?

   Answer: **Immutable types:** Cannot be changed after creation. Examples: `int`, `float`, `str`, `tuple`.

   ```
   x = "hello"

   x[0] = "H"   # Error, strings are immutable
   ```

   Mutable types: Can be modified after creation. Examples: `list`, `dict`, `set`.

   ```
   mylist = [1, 2, 3]

   mylist[0] = 100   # Allowed, lists are mutable
   ```

Importance in Data Science:

- Mutability affects **memory management and performance**.
- Immutable objects (like strings, tuples) are safer for keys in dictionaries and ensure data integrity.
- Mutable objects (like lists, DataFrames) allow flexible manipulation of datasets, which is essential for preprocessing and analysis.