2(a) Difference between **is** and **==** in Python:

is checks identity (whether two variables point to the same object in memory).

== checks equality of values (whether the values are the same).

Example:

x = [1,2,3]; y = x; z = [1,2,3]

x is y → True (same object)

x == z → True (same value)

x is z → False (different objects)

2(b) How zip() works:

zip() combines elements from multiple iterables (like lists/tuples) into tuples, stopping at the shortest iterable.

Example:

list(zip([1,2,3], ['a','b','c'])) → [(1,'a'), (2,'b'), (3,'c')]

2(c) Difference between loc[] and iloc[] in pandas:

loc[] – label-based indexing (uses row/column names)

iloc[] – position-based indexing (uses integer positions)

Example:

df.loc[1, 'Name'] # row label 1, column 'Name'

df.iloc[1, 0] # 2nd row, 1st column

2(d) Broadcasting in NumPy:

Broadcasting allows operations between arrays of different shapes by expanding smaller arrays.

Example:

[1,2,3] + 5 → [6,7,8]

2(e) any() vs all():

any() → True if at least one element is True.

all() → True only if all elements are True.

Example:

any([0, False, 3]) → True

all([1, 2, 3]) → True

22(a) Count occurrences of unique values in pandas Series:

Use value_counts().

Example:

s = pd.Series([1,2,2,3])

s.value_counts() → 2:1, 2:2, 3:1

22(b) Difference between axis=0 and axis=1 in pandas:
axis=0 → operate column-wise (down rows)
axis=1 → operate row-wise (across columns)

22(c) reshape() vs resize() in NumPy:
reshape(): returns a new array, original remains unchanged (if possible).
resize(): changes the original array in-place, may fill/trim data.

22(d) Lazy evaluation in map() and filter():
map() and filter() return iterators; values are produced only when iterated, not immediately computed.

22(e) Merging two pandas DataFrames:
Use merge() on a common column.
Example:
pd.merge(df1, df2, on='id')