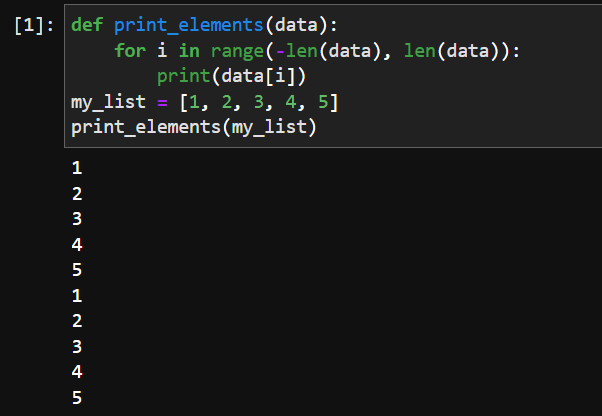
Q1. Can you create a programme or function that employs both positive and negative indexing? Is there any repercussion if you do so?

🡪 

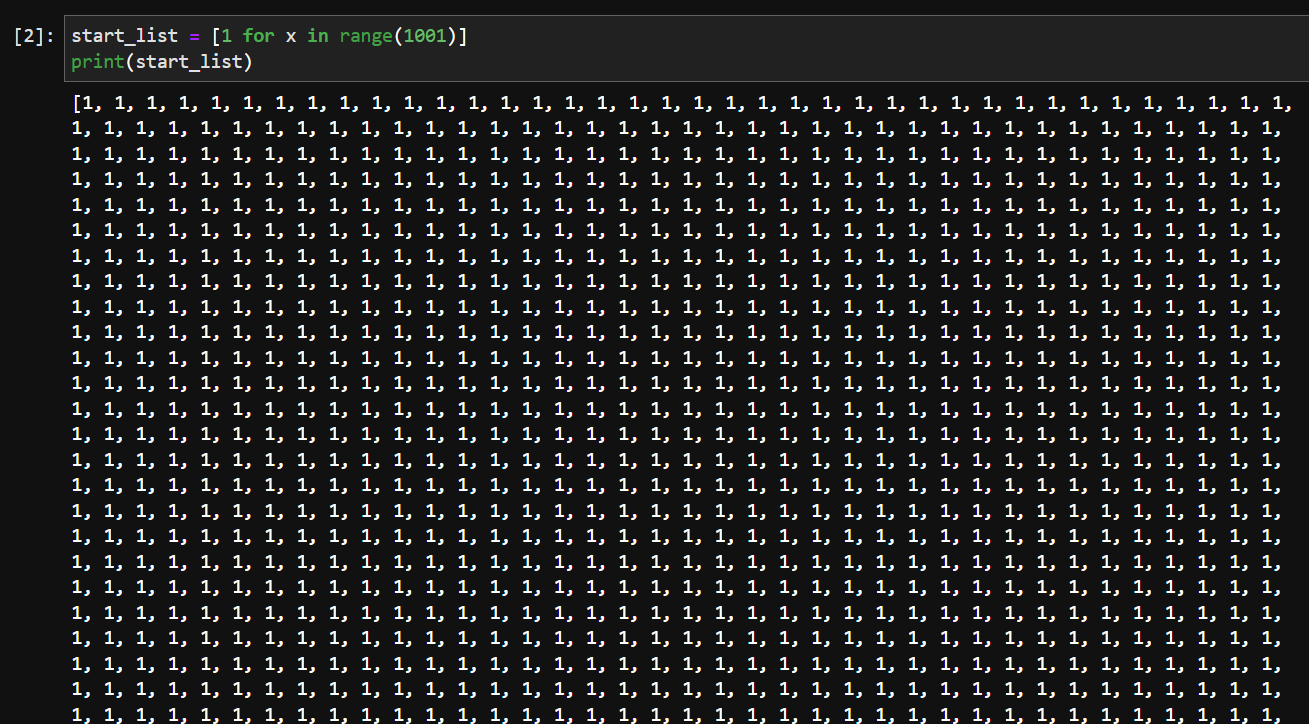
However, there can be repercussions if you mix positive and negative indices carelessly. It can lead to confusion and potential indexing errors in your program. Here are a few points to consider:

Indexing Consistency: Mixing positive and negative indices might make your code less readable and harder to maintain. It's generally recommended to stick to one indexing convention (either positive or negative) to ensure consistency.

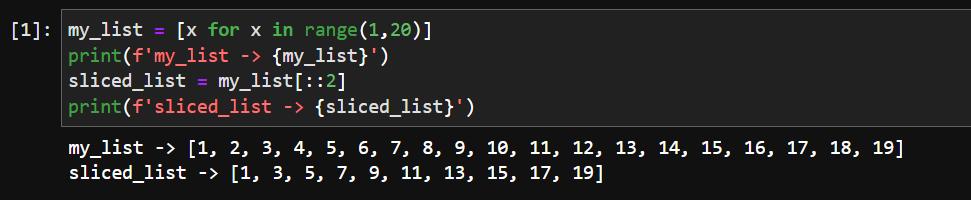
Off-by-One Errors: When using mixed indexing, you need to be cautious about the range of indices you iterate over. Negative indices start from -1, whereas positive indices start from 0. This discrepancy can lead to off-by-one errors if you're not careful.

Semantic Clarity: Mixing positive and negative indices might make the intent of your code less clear. It's generally better to use positive indices for accessing elements from the start of a sequence and negative indices for accessing elements from the end.

Q2. What is the most effective way of starting with 1,000 elements in a Python list? Assume that all elements should be set to the same value.

🡪 

Q3. How do you slice a list to get any other part while missing the rest? (For example, suppose you want to make a new list with the elements first, third, fifth, seventh, and so on.)

🡪 

Q4. Explain the distinctions between indexing and slicing.

🡪 Indexing is used when we have to work on index level. While slicing are used over a range of items.

Q5. What happens if one of the slicing expression's indexes is out of range?

🡪 If start index is out of range then it will return empty entity.

Q6. If you pass a list to a function, and if you want the function to be able to change the values of the list—so that the list is different after the function returns—what action should you avoid?

🡪 Always use return statement, if we want the see the changes in the input list.

Q7. What is the concept of an unbalanced matrix?

🡪 In Unbalanced Matrix number of rows is not same as number of columns.

Q8. Why is it necessary to use either list comprehension or a loop to create arbitrarily large matrices?

🡪 List comprehension or a Loop helps creation of large matrices easy. it also helps to implemeent and avoid manual errors. it also makes reading code easy. Also lot of time for manual feeding is reduced.