(b) [4 marks] **Function definitions.** The following code contains an incomplete function definition. The missing parts are indicated by ellipses (...). Read this definition and then answer the questions below.

```
def mystery(...) -> ...:
    """...

    Doctest example:

    ...
    """
    if str.upper(names[index]) == 'CSC':
        return 'Found'
    else:
        return 'Not found: ' + message
```

(i) What is the *name* of the function being defined?

mystery

(ii) What is the *return type* of the function?

It returns a string or an error would appear while adding variable message to a string.

(iii) What are the *parameter names* and what is the *type* of each parameter? For a collection type (`set`/`list`/`dict`), assume the collection is homogeneous and specify the contained type as well.
If there is more than one option for a type annotation, pick one and briefly justify your choice.

names : list [str] → because it is being indexed and string methods are being called on its elements.
index : int → used to index a list
message : str → added to a string

(iv) Write *one* possible doctest example for this function. If the function takes multiple parameters, you may choose the order in your example.

```
>>> mystery(names=['CSC', 110, 'David'], index = 0, message = 'no CSC course')
'Found'
```

(v) In addition to the parameter types, this function requires another *precondition* to ensure its body does not raise an error. Write that precondition as a single Python expression (possibly using `and`/`or`).

names != [] and 0 <= index < len(names) and type(message) == class <str>