

- If you have any scratch work, please circle your final answer.
- Any code you write should run in a Jupyter cell; every character counts!
- For all questions on this exam, assume that all necessary packages have been imported.

1. In the following problems, write what each code block would display if executed in a Jupyter cell. If it would display an error or create an infinite loop, write “error” or “infinite loop”.

- (a) `name = 'appm1650'`
`name[::-2] + 2 * name[:3]`
- (b) `data = ['apple', 12, (3.0, 4.0)]`
`(isinstance(data[2], float)) or (data[1] // 5 == 2)`
- (c) `nums = [2 * n for n in range(4)]`
`nums[1] // 2, nums[3] % nums[2]`
- (d) `pairs = zip(range(3), range(1, 4))`
`inc = 0`
`for a, b in pairs:`
 `print(a, b)`
 `inc += a * b`
`inc`

Solution:

- (a) `'06mpappapp'`
- (b) `True`
- (c) `(1, 2)`
- (d) `0 1`
`1 2`
`2 3`
`8`

2. Write a function called `string_editor(val, old, new)` which takes the string `val` and returns the same string but where the single character specified by `old` is replaced by the single character `new`. For example, the code `string_editor('appm1650', 'p', 'L')` would return `'aLLm1650'`.

Solution:

```
def string_editor(val, old, new):
    output = ''
    for v in val:
        if v == old:
            add = new
        else:
            add = v
        output += add
    return output
```

3. University instructors must submit “course alerts” for students with low grades. Write a function `alerts(roster, cutoff)`, which takes a list of name/grade tuples given by `roster`, and returns a list of names of students whose grades fall below the float `cutoff`. You may assume that student names are strings, and their grades are floats between 0.0 and 1.0.

For example, if `roster = [('kate', 0.62), ('fred', 0.55), ('mona', 0.89)]`, then `alerts(roster, 0.68)` would return the list `['kate', 'fred']` because Kate and Fred’s scores are below 0.68.

Solution:

```
# using a list comprehension:
def alerts(roster, cutoff):
    return [name[0] for name in roster if name[1] < cutoff]

# using a for-loop:
def alerts(roster, cutoff):
    names = []
    for name, grade in roster:
        if grade < cutoff:
            names.append(name)

    return names
```

4. An *arithmetic sequence* is a sequence of numbers where the difference between adjacent terms is constant. For example, the list

```
good_nums = [2, 5, 8, 11]
```

forms an arithmetic sequence because the difference between subsequent and previous terms is always 3. The list

```
bad_nums = [1, 4, -5, 0]
```

does not form an arithmetic sequence because the differences between terms is not constant.

Write a function `is_arithmetic(nums)` which returns `True` if the numbers in `nums` form an arithmetic sequence, and `False` otherwise. You may assume that the values in `nums` are integers, and that it contains at least two values.

Solution:

```
def is_arithmetic(nums):  
    diff = nums[1] - nums[0]  
    for i in range(2, len(nums)):  
        if diff != nums[i] - nums[i - 1]:  
            return False  
    return True
```