- 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". In all of these problems, the object `arr` is given by

   ```
   arr = np.arange(1, 7).reshape(3, 2)
   ```

   (a)
   ```
   arr.flatten().astype(float)
   ```

   (b)
   ```
   arr[2, :] - 1
   ```

   (c)
   ```
   arr[::2, 1:] = 0
   arr
   ```

   (d)
   ```
   print(arr ** 2 == arr + 2)
   arr[arr ** 2 == arr + 2]
   ```

2. (a) Write a function `factors(n)` which returns a list of all positive factors of `n`. For example, `factors(6)` would return `[1, 2, 3, 6]`.

   (b) Write a function `most_factors(n)` which returns the number between 1 and n (inclusive) which has the most positive factors. If multiple numbers have the same number of factors, `most_factors(n)` should return the largest one.

3. (a) Write a function `encode(word, shuffled)` which encodes the letters in a word by replacing them with the corresponding letter in shuffled. To give a simplified example, suppose that the alphabet consists of the characters `'abc'`, and `shuffled='bac'`. Then `encode('cab', shuffled)` would return `'cba'`, because `shuffled` swaps the letters `'a'` and `'b'`.

   You may fill in code here, if you prefer:

```
def encode(word, shuffled):
    alphabet = 'abcdefghijklmnopqrstuvwxyz'
```

   (b) Suppose you want to *decode* an output from the function `encode()`, to get the original word back. How would you do that? You may either explain in words what you would change about the function `encode()`, or you may also write the code.

4. Consider the weather data below, indexed by days since a last April 19.

	Temp	Clouds
0	46	True
1	56	False
2	62	True
⋮	⋮	⋮

   (a) Add today's weather records: the temperature is 65, and there are clouds.

   (b) Create a column `'Diff'` which records the difference between the daily temperature and the overall mean temperature in the dataset.

   (c) Among all cloudy days, write code which returns a list of the five indices that correspond to the largest difference from the mean temperature.

5. Create a class called `Animal`. Each instance has the following attributes to record information about that type of animal:

   - `name`: the name of the animal,
   - `n_eyes`: the number of eyes the animal has,
   - `n_limbs`: the number of limbs (arms or legs) the animal has,
   - `digits_per_limb`: the number of digits (fingers or toes) per limb.

   And the methods:

   - `total_digits()`: returns the total number of digits an animal has
   - `more_eyes(animal)`: compares the number of eyes between the current instance and the object `animal`, and returns the name of the animal that has more eyes (you may assume one does have more)
   - `summary()`: prints a formatted short summary of the animal, and returns nothing. For example, `human.summary()` would print

     ```
     human
     2 eyes
     4 limbs
     ```

More space on the next page (you may break your code up between pages).

6. Write a function `plot_row_sum(arr)` which plots the **sums** of the rows in the Numpy array `arr` as a scatterplot. If the entries of the array are integers, the dots should be red; if the entries are floats, they dots should be blue. Use **try except** to catch invalid input, in which case no plot is created; instead, display the message "invalid input".