CS 114
Final review

Nov 28 2025

Final Topics

- All Midterm Topics
- Module 8: Masking
- Module 9: Classes
- Module 10: Recursion
- Module 11: Efficiency
- Style (docstrings, annotations, asserts)

Final date: Thursday Dec. 11, 7:30pm-10:00pm

Module 10: Recursion

- A recursive function is a function that calls itself.
- It needs a base case to stop, or it will run forever.
- Each call usually works on a smaller part of the problem that approaches the base case.
- The results from smaller calls are combined to get the final answer.

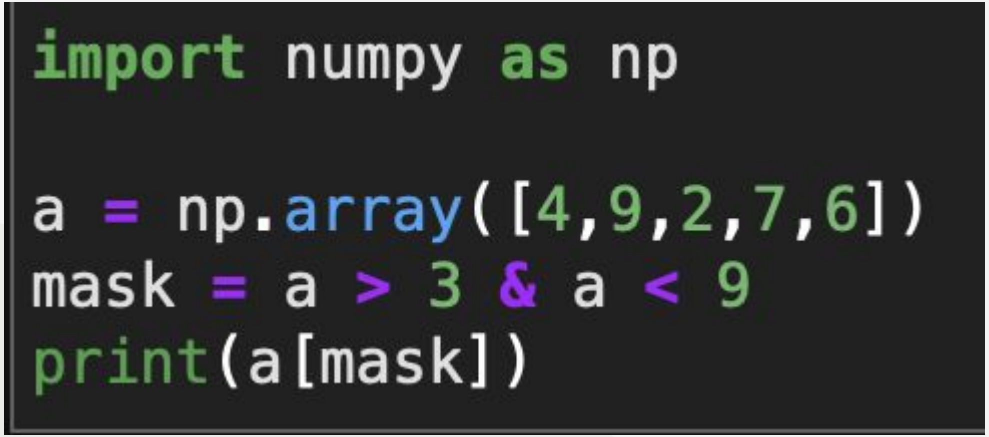
Write a recursive function count_odd(lst) that counts the number of **odd** integers in the list lst. You can only use recursion, no loops.

Solutions found in Jupyter Notebook

Go to vevox.com

Sign in using the session ID: 149-788-037

Masking



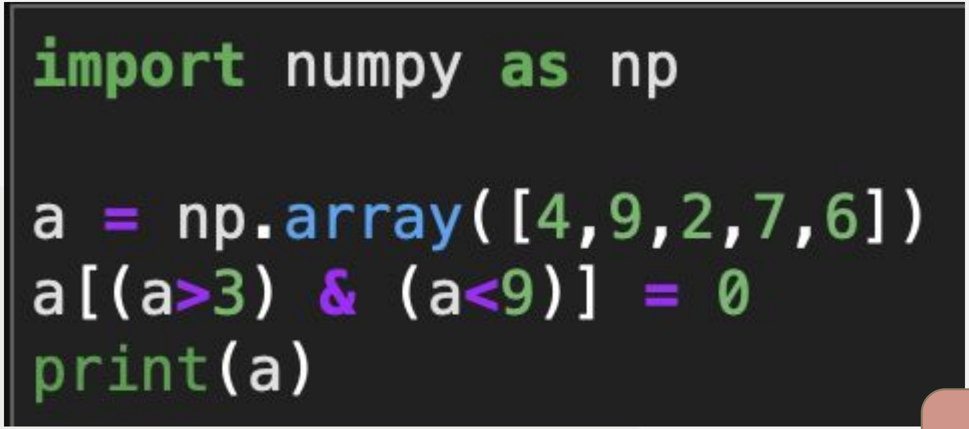
```python
import numpy as np

a = np.array([4,9,2,7,6])
mask = a > 3 & a < 9
print(a[mask])
```

Missing brackets around the mask conditions

What does this code print?

A. [4,9,2,7,6]   B. [4,5,7,6]   C. [4,7,6]   [✓ D. error]

Left Hand Masking



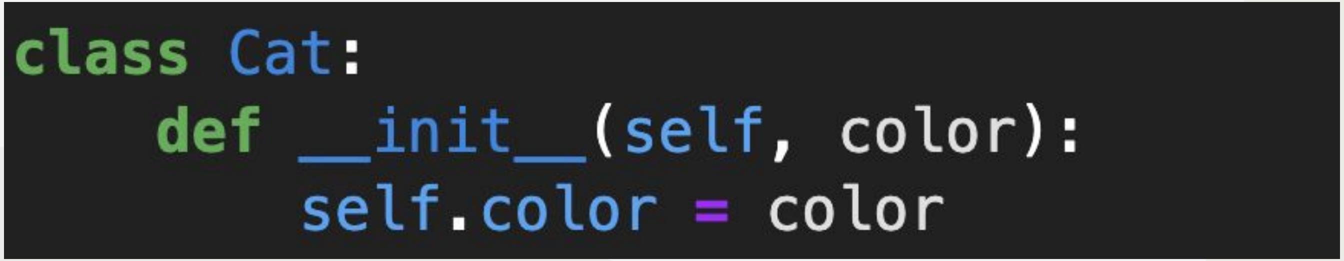
```
import numpy as np

a = np.array([4,9,2,7,6])
a[(a>3) & (a<9)] = 0
print(a)
```

Where the mask is True, values will be replaced with 0

What does this code print?

A. [4,7,6]   B. [4,0,0,7,6]   [✓ C. [0,9,2,0,0]]   D. error

Classes



```
class Cat:
    def __init__(self, color):
        self.color = color
```

Which of the following is to create a object of class Cat?

A.c = Cat.color("Black")

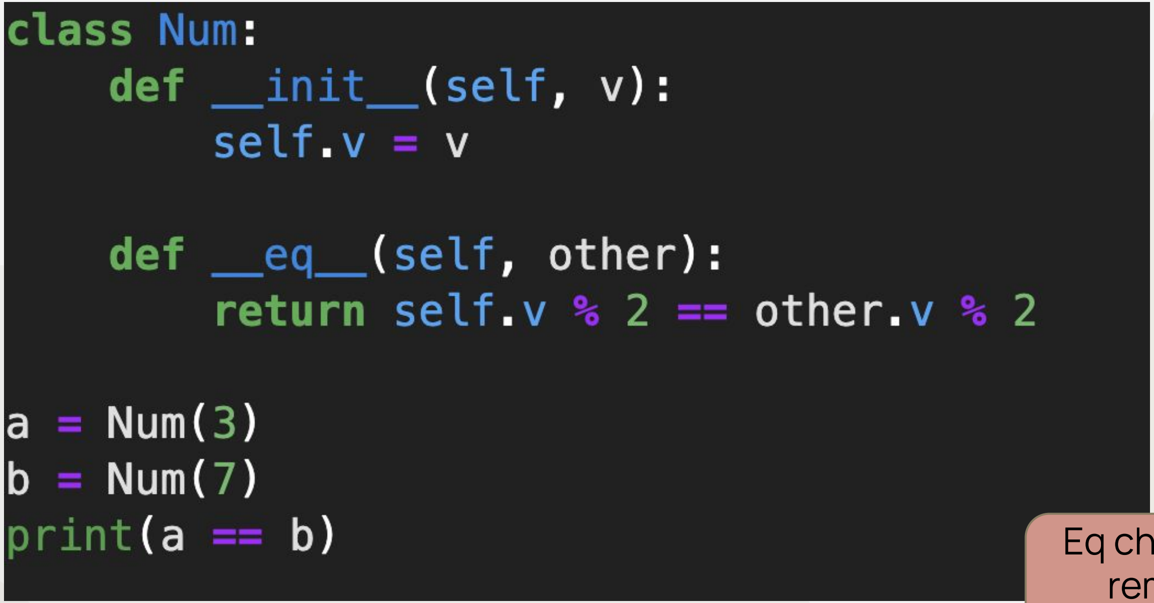
B.c = Cat("Black")

C.c = Cat.self("Black")

D.c = Cat.new("Black")

To create a class object call the class with needed parameters.

Special Method Classes



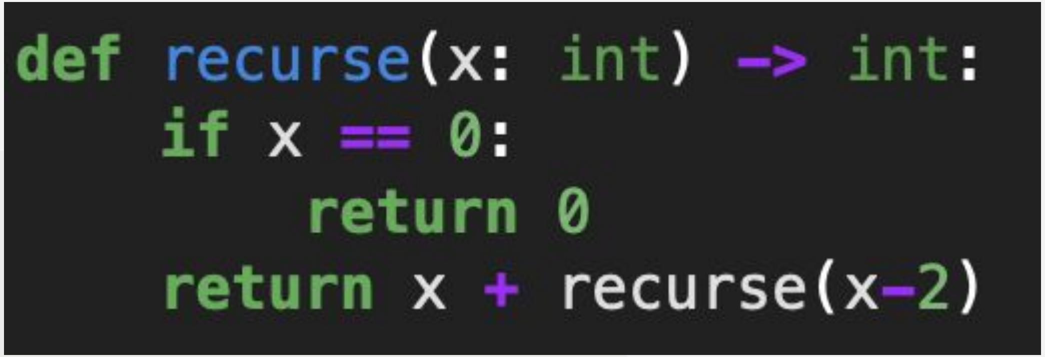
```python
class Num:
    def __init__(self, v):
        self.v = v

    def __eq__(self, other):
        return self.v % 2 == other.v % 2

a = Num(3)
b = Num(7)
print(a == b)
```

Eq check whether the remainders after division by 2 are equal

What does this print?

A. False [✓ B. True] C. Error: cannot compare objects

Recursion



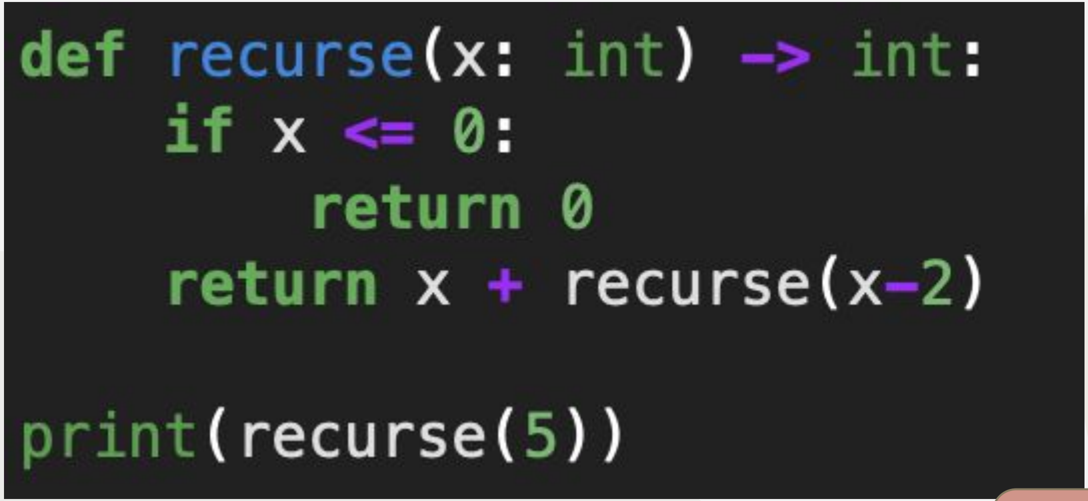
```
def recurse(x: int) -> int:
    if x == 0:
        return 0
    return x + recurse(x-2)
```

Negative values and odd numbers will run forever since -2 will skip the == 0 condition.

Does this code run in an infinite loop?

A. Yes B. No [✓ C. Sometimes]

Recursion



```
def recurse(x: int) -> int:
    if x <= 0:
        return 0
    return x + recurse(x-2)

print(recurse(5))
```

Adds 5 + 3 + 1 + 0

What does this code print?

A. 15 [✓ B. 9] C. 8 D. 4

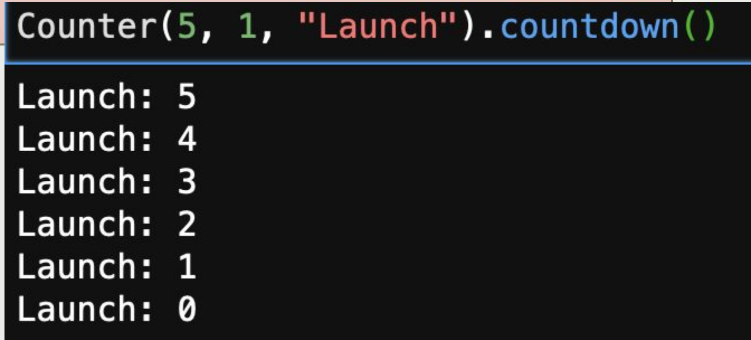
```
!wget https://student.cs.uwaterloo.ca/~cs114/src/plot_me.csv
```

CSV Files, Plotting, & Masking

Download the CSV file above and write a code that will take open a csv with columns called “X” and “Y”. You will then plot this data however it must be masked so that the X values only range from 0.0-10.0.

Recursion & Classes:

Create a class called Counter that stores the attributes value, step, and message. The first two are ints, message is a string. Write a method called countdown() that will recursively reduce the value by the step until it reaches a value less than 0 and print each value as it goes.



```
Counter(5, 1, "Launch").countdown()
```

```
Launch: 5
Launch: 4
Launch: 3
Launch: 2
Launch: 1
Launch: 0
```

Solutions found in Jupyter Notebook