

CSc 110

Sort + Recursion

Adriana Picoral (she/her/hers)

Announcements


- Final exam: Friday (December 9) at 1pm (2 hours)
- Review session: 5pm on Thursday December 8th
 - Room TBA
- How to study for the exam? Study guide, textbook, slides

Recursion

- A recursive function calls itself
- The idea is to solve a smaller subproblem of the original problem
- Case (conditional) needs to be provided to terminate the recursion process

What will these print?

```
def sequence(n):
```



```
    print(n)
```

```
    if n > 0:
```

```
        sequence(n-1)
```

```
def main():
```

```
    sequence(9)
```


```
main()
```

X

```
def sequence(n):
```

```
    if n > 0:
```

```
        sequence(n-1)
```



```
    print(n)
```

```
def main():
```

```
    sequence(9)
```

```
main()
```

What do you need to change to get the output?

```
def sequence(n):
```

```
    print(n)
```

```
    if n > 0:
```

```
        sequence(n-1)
```

```
def main():
```

```
    sequence(9)
```

```
main()
```

9
8
7
6
5
4
3
2
1
0



10
9
8
7
6
5
4
3
2
1

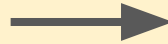
What do you need to change to build a list of numbers instead of printing them?

```
def sequence(n) :  
    print(n)  
    if n > 1:  
        sequence(n-1)
```

```
def main() :  
    sequence(10)
```

```
main()
```

10
9
8
7
6
5
4
3
2
1



[10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

Factorial

Factorial of a number is the product of all the integers from 1 to that number. For example, the factorial of 6 (denoted as 6!) is $1*2*3*4*5*6 = 720$

Write the code: Recursive Factorial

Factorial of a number is the product of all the integers from 1 to that number. For example, the factorial of 6 (denoted as 6!) is $1*2*3*4*5*6 = 720$

We know that factorial of 1 is 1

The rest is of the function is n multiplied by factorial of n-1

Merge Sort

- Find the middle point to divide the list into two halves
- Call function for first half
- Call function for second half
- Merge the two halves sorted

Lots of algorithms

- There are many sorting algorithms
 - **Bogo sort**
 - **Selection sort**
 - **Bubble sort**
 - **Insertion sort**
 - **Merge sort**
 - **Quick sort**
 - ...more...

Timing and Vis

- `sort_timing.py`
- <https://www.youtube.com/watch?v=kPRA0W1kECg>

More recursion: write a recursive function to check if a word is a palindrome

```
is_palindrome("banana") # False  
is_palindrome("racecar") # True  
is_palindrome("python") # False  
is_palindrome("civic") # True  
is_palindrome("hannah") # True
```

try/except

- Can be used to catch errors in a program without it having to crash!
- The programmer can provide more information and/or context as to what happened
- <https://docs.python.org/3/library/exceptions.html>

What will happen?

```
ages = [35, 35, 23, 18, 45, 18, 72]
```

```
try:
```

```
    print(ages[0] + ages[4])
```

```
except:
```

```
    print('Failed to print the sum of two ages')
```

What will happen?

```
ages = [35, 35, 23, 18, 45, 18, 72]
try:
    for i in range(len(ages)):
        if ages[i] > ages[i+1]:
            print(ages[i])
except:
    print('Issue with comparison')
```

What will happen?

```
ages = [35, 35, 23, 18, 45, 18, 72]
for i in range(len(ages)):
    try:
        if ages[i] > ages[i+i]:
            print(ages[i])
    except:
        print('Issue with comparison')
print(ages[-1])
```


Statement A

try:

Statement B

Statement C

except:

Statement D

Statement E

Statement F

What will happen?

```
ages = [35, 35, 23, 18, 45, 18, 72]
for i in range(len(ages)):
    try:
        if ages[i] > ages[i+i]:
            print(ages[i])
    except Exception as e:
        print(e)
```

Try / Except

- Can make debugging easier if you provide clear error messages
- Can more clearly explain legitimate issues
 - For example: poorly formatted input file, user input issue