15-112 Fundamentals of Programming

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Life Lesson

Loosen up. Relax.

Except for rare life-and-death matters, nothing is as important as it first seems

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Announcements

- ❑Midterm on Thursday – October 3.
- ❑Assignment 5 Due Tuesday October 8

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

What are we doing today?

- ❑Recursion

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Recursion

- A method of simplification that involves dividing a problem into simpler subproblems of the same type.
- A function that calls itself is called a recursive functions

Factorial Example

```
def factorial(n):
    if n == 0:
        return 1
    return n * factorial(n-1)
```

Base Case

Recursive Case

General Form of Recursive Functions

Function (input)

- base case – this will stop chain of calls
- recursive case – call Function with simpler input

Definition of Factorial

F(0) = 1

F(n) = n*F(n-1)

Definition of power

P(b,0) = 1

P(b,e) = b * P(b,e-1)

Recursive Power

```
def power(base, exp):
    if (exp == 0):
        return 1
    else:
        return base * power(base, exp-1)
```

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Sum of All elements in a list

F([ ]) = 0

F([x,….]) = x + F([….])

Sum of All elements in a list

```
def listSum(list):
    if (len(list) == 0):
        return 0
    else:
        return list[0] + listSum(list[1:])
```

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Exercise

- We want to write a function that counts how many odd numbers exist in a list
- Write the function definition
- Write the recursive function

Fibonacci

- Following is the Fibonacci series

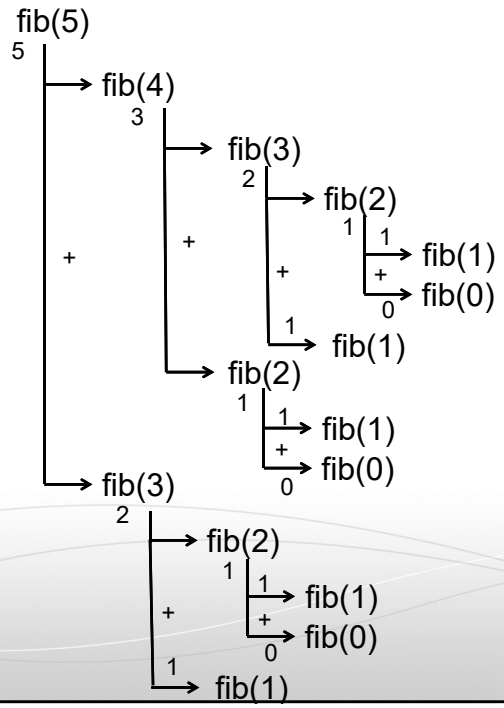
  0, 1, 1, 2, 3, 5, 8, 13, 21, 34
- In general

  $F_n = F_{n-1} + F_{n-2}$

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Recursive Function

```
def fib(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fib(n-1) + fib(n-2)
```

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Palindromes

- A palindrome is a word or sentence that reads the same both forwards and backwards
  - rotor
  - madam
  - rats live on no evil star

Recursive palindromes

- How would describe a palindrome recursively?

F(n) = true if n == “”
true if len(n) == 1
F(n[1→len-1] if F[0] == F[-1]
false if F[0] != F[-1]

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Recursive palindrome

```
def palindrome(a):
    if a == "":
        return True
    elif len(a) == 1:
        return True
    elif a[0] == a[len(a)-1]:
        return palindrome(a[1:len(a)-1])
    else:
        return False
```


Do Parenthesis Match?

- "3*(8-4) / (4 – 9)"
- "3*(((8*2)-4) / (4 – 9))"
- "3*(((8*(2+9)))-4)(/ ((11*4) – 9)))"

جامعة كارنيجي ميلون في قطر
Carnegie Mellon Qatar

Towers of Hanoi

```
def move(n, frm, to, via):
    if (n == 1):
        print (frm, to)
    else:
        move(n-1, frm, via, to)
        move(  1, frm, to,  via)
        move(n-1, via, to,  frm)
```


Another Exercise

We have bunnies standing in a line, numbered 1, 2, ... The odd bunnies (1, 3, ..) have the normal 2 ears. The even bunnies (2, 4, ..) we'll say have 3 ears, because they each have a raised foot. Recursively return the number of "ears" in the bunny line,

bunnyEars2(0) → 0
bunnyEars2(1) → 2
bunnyEars2(2) → 5