

## Lecture 5: July 21

*Lecturer: Anwar Mamat*

**Disclaimer:** *These notes may be distributed outside this class only with the permission of the Instructor.*

## 5.1 Recursion

### 5.1.1 Fibonacci Numbers

Listing 1: Fibonacci Numbers

```
1 #bad solution
2 def fib(n):
3     print('inside_recursive_call')
4     if(n < 1):
5         print('error_message')
6         return 0
7     if(n==1 or n == 2):
8         return 1
9     return fib(n-1)+fib(n-2)
10
11 #iterative solution
12 def fib_iter(n):
13     if(n == 1 or n == 2):
14         return 1
15     a = 1
16     b = 1
17     i = 1
18     while( i < n):
19         a,b=b,a+b
20         i += 1
21     return a
22
23 def main():
24     print(fib_iter(5))
25
26 main()
```

### 5.1.2 GCD: Greatest Common Divisor

The Euclidean algorithm is a method for computing the greatest common divisor (GCD).

$$gcd(a, b) = \begin{cases} a & \text{if } b = 0 \\ gcd(b, a\%b) & \text{otherwise} \end{cases}$$

Listing 2: Greatest Common Divisor

```

1 #recursive version
2 def gcd(a,b):
3     if(b == 0):
4         return a
5     else:
6         return gcd(b, a%b)
7 #iterative version
8 def gcd_iter(a,b):
9     while(b != 0):
10        a,b=b,a%b
11    return a
12 def main():
13    a = 1100
14    b = 13176
15    if(b > a):
16        a,b = b,a # if b >a, swap a,b
17    c = gcd(a,b)
18    print(c)
19 main()

```

### 5.1.3 Calculate $1 + 2 + 3 + \dots + n$

Listing 3: Sum of Numbers

```

1 def sum(n):
2     if(n == 1):
3         return 1
4     else:
5         return n + sum(n-1)
6 #iterative function
7 def sum_iter(n):
8     sum = 0
9     while(n >= 1):
10        sum = sum + n
11        n = n -1
12    return sum
13 def main():
14    t = 10
15    s = sum_iter(t)
16    print('Sum({})={}'.format(t,s))
17
18 main()

```

### 5.1.4 Palindrome

A palindrome is a word, phrase, number, or other sequence of symbols or elements that reads the same forward or reversed, with general allowances for adjustments to punctuation and word dividers. Famous

examples include "Amor, Roma", "A man, a plan, a canal: Panama", "race car"

Listing 4: Palindrome

```

1 #recursive version
2 def palindrome(input_string):
3     l = len(input_string)
4     if(l == 0 or l == 1):
5         return True
6     if(input_string[0] != input_string[l-1]):
7         return False
8     return palindrome(input_string[1:l-1])
9 #iterative version
10 def pal_iter(input_string):
11     l = len(input_string)
12     s = 0
13     t = l - 1
14     while(s <= t):
15         if(input_string[s] != input_string[t]):
16             return False
17         s += 1
18         t -= 1
19     return True
20 #another version
21 def pal2(input_string):
22     x = len(input_string)
23     y = 0
24     for i in range(x//2):
25         if(input_string[i] != input_string[x-1-i]):
26             return False
27     return True
28
29 def main():
30     s = "A_____but_tuba."
31     s = s.lower()
32     #filter punctuation marks
33     s = s.replace("_", "")
34     s = s.replace("%", "")
35     s = s.replace(", ", "")
36     s = s.replace(".", "")
37     r = pal2(s)
38     if(r):
39         print('{}_is_a_palindrome'.format(s))
40     else:
41         print('{}_is_NOT_a_palindrome'.format(s))
42
43 main()

```

### 5.1.5 Regular Expressions

- import re

- regular expression search and match
- patterns