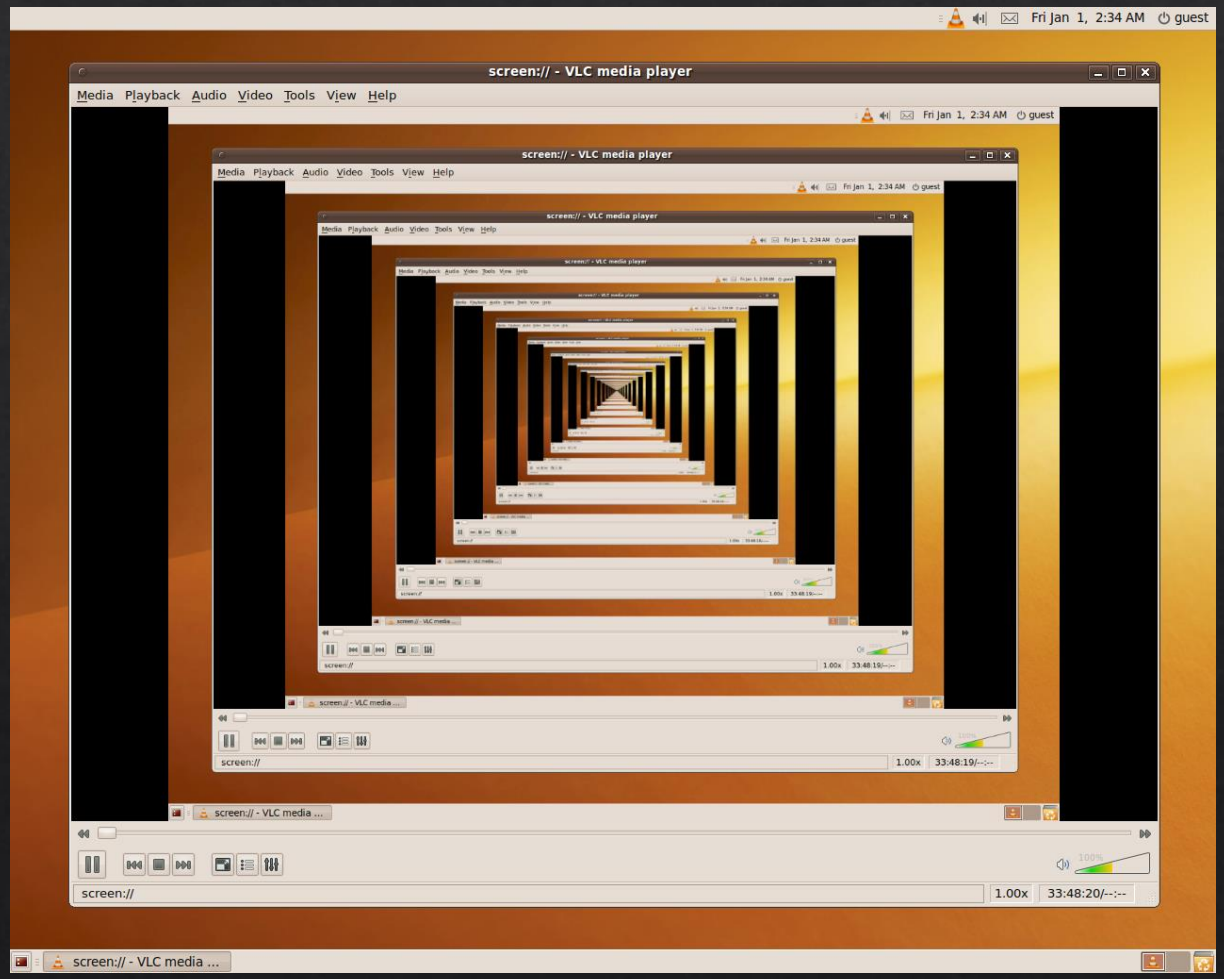
PPL: Lab 3

Recursion - רקורסיה

Recursion - רקורסיה

הגדרה

רֶקוּרְסִיָּה (בעברית: נסיגה) היא תופעה שכל מופע שלה מכיל מופע נוסף שלה, כך שהיא מתרחשת ומשתקפת בשלמותה בתוך עצמה שוב ושוב.



פונקציה רקורסיבית

```
def f1():
    print('start f1 !')
    f1()
    print('end f1 !')
f1()
```

```
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
start f1 !
```

פונקציה רקורסיבית

```
def f2(x):
    if x > 0:
        print('start f2 !')
        f2(x-1)
        print('end f2 !')
f2(5)
```

```
start f2 !
start f2 !
start f2 !
start f2 !
start f2 !
end f2 !
end f2 !
end f2 !
end f2 !
end f2 !
```

פונקציה רקורסיבית

```
def f5(x):
    if x > 0:
        print(x, end=' ')
        f5(x-1)
        print(x, end=' ')
```

```
>>> f5(7)
7 6 5 4 3 2 1 1 2 3 4 5 6 7
```

פונקציה רקורסיבית

```
def f6(x):
    if x == 1:
        print(x, end=' ')
    else:
        print(x, end=' ')
        f6(x-1)
        print(x, end=' ')
```

```
>>> f6(7)
7 6 5 4 3 2 1 2 3 4 5 6 7
```

פונקציה רקורסיבית

```
def f7(x,y=1):
    if x == y:
        print(y, end=' ')
    else:
        print(y, end=' ')
        f7(x,y+1)
        print(y, end=' ')
```

```
>>> f7(7)
1 2 3 4 5 6 7 6 5 4 3 2 1
```

פונקציה רקורסיבית

```
def f8(x):
    if x == 0:
        return 1
    return x*f8(x-1)
```

```
>>> f8(6)
720
>>> f8(0)
1
>>> f8(11)
39916800
```

Python Bitwise Operators

Operator	Description	Example(a=60(111100), b=13(1101))
& - Binary AND	Operator copies a bit to the result if it exists in both operands	(a & b) (means 0000 1100)
| - Binary OR	It copies a bit if it exists in either operand.	(a | b) = 61 (means 0011 1101)
^ - Binary XOR	**It copies the bit if it is set in one operand but not both.**	**(a ^ b) = 49 (means 0011 0001)**
~ - Binary Ones Complement	It is unary and has the effect of 'flipping' bits.	(~a) = -61 (means 1100 0011 in 2's complement form due to a signed binary number.
<< - Binary Left Shift	The left operands value is moved left by the number of bits specified by the right operand.	a << 2 = 240 (means 1111 0000)
>> - Binary Right Shift	The left operands value is moved right by the number of bits specified by the right operand.	a >> 2 = 15 (means 0000 1111)

פונקציה רקורסיבית

```
def f9(x):
    if x == 0:
        return 1
    return (x%2)^f9(x//2)
```

```
>>> f9(15)
1
>>> f9(131)
0
```

פונקציה רקורסיבית

```
def exp(x, n):
    if n == 0:
        return 1
    return x * exp(x, n-1)
```

```
exp(2, 4)
+-- 2 * exp(2, 3)
|       +-- 2 * exp(2, 2)
|       |       +-- 2 * exp(2, 1)
|       |       |       +-- 2 * exp(2, 0)
|       |       |       |       +-- 1
|       |       |       +-- 2 * 1
|       |       |       +-- 2
|       |       +-- 2 * 2
|       |       +-- 4
|       +-- 2 * 4
|       +-- 8
+-- 2 * 8
+-- 16
```

פונקציה רקורסיבית

```python
def fast_exp(x, n):
    if n == 0:
        return 1
    if n%2 == 0:
        return fast_exp(x*x, n//2)
    return x * fast_exp(x, n-1)
```

```
fast_exp(2, 10)
+-- fast_exp(4, 5) # 4 = 2 * 2
|   +-- 4 * fast_exp(4, 4)
|   |       +-- fast_exp(16, 2) # 16 = 4 * 4
|   |       |   +-- fast_exp(256, 1) # 256 = 16 * 16
|   |       |   |   +-- 256 * fast_exp(256, 0)
|   |       |   |             +-- 1
|   |       |   |   +-- 256 * 1
|   |       |   |   +-- 256
|   |       |   +-- 256
|   |       +-- 256
|   +-- 4 * 256
|   +-- 1024
+-- 1024
1024
```