

Outline

- 1 midterm exam on Friday 11 July 2014
 - policies for the second part
- 2 some review questions
 - with answers

MCS 260 Lecture 10.5
Introduction to Computer Science
Jan Verschelde, 9 July 2014

review of the first 9 lectures

- 1 midterm exam on Friday 11 July 2014
 - policies for the second part
- 2 some review questions
 - with answers

policies

The second part of the exam will be open notes, books, and with a computer.

The material in this part has its focus on Python:

- 1 variables and assignments, elementary and composite types, intrinsic operations, tuples, lists, dictionaries, anydbm, queues and stacks;
- 2 if else elif, while and for, queues and stacks, top down design, functions, lambda, list comprehensions.

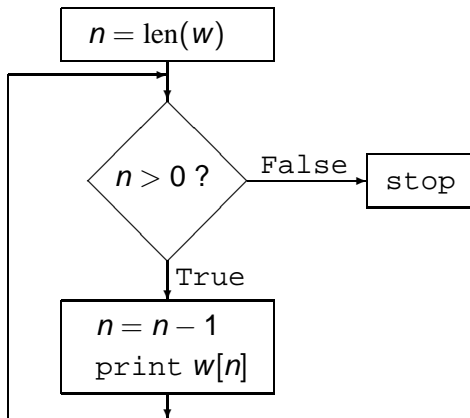
This review contains some preliminary examples of questions which may help you prepare for the second part of the midterm exam.

review of the first 9 lectures

- 1 midterm exam on Friday 11 July 2014
 - policies for the second part
- 2 some review questions
 - with answers

question 1

Consider the flowchart, that takes on input the word w :



- 1 Write Python code based using the same control logic as shown in the flowchart.
- 2 Write Python code that uses a for loop.

answer to question 1

```
WORD = raw_input('give a word : ')
N = len(WORD)
while N>0:
    N = N - 1
    print WORD[N]

N = len(WORD)
for k in range(N):
    print WORD[N-k-1]
```

question 2

Consider the following pseudocode:

```
input: x, a floating point number;
       c, a list of coefficients.
output:  $c[0] + c[1]*x + c[2]*x**2 + \dots + c[n]*x**n$ ,
        where  $n = \text{len}(c) - 1$ .
result = c[0]; i = 0; n = len(c); p = x
as long as i < n-1 do
    result = result + c[i+1]*p
    p = p * x; i = i + 1
output result
```

Write a Python script that implements this pseudocode.

answer to question 2

```
CRAW = raw_input('give a list of coefficients : ')
C = eval(CRAW)
XRAW = raw_input('give a floating points number : ')
X = float(XRAW)
RESULT = C[0]
i = 0
N = len(C)
P = X
while i < N-1:
    RESULT = RESULT + C[i+1]*P
    P = P * X
    i = i + 1
print RESULT
```

question 3

What data structure of Python would you use to store the derivatives of trigonometric expressions like \cos , \sin , and \tan ? Illustrate with actual Python code.

answer to question 3

```
D = {'cos': '-sin', 'sin': 'cos', 'tan': 'tan**2 + 1'}
for k in D.keys():
    print 'the derivative of', k, 'is', D[k]
```

question 4

Give the Python code to generate a list of all tuples containing all possible input values for a Boolean expression, i.e.: [(False,False),(False,True),(True,False),(True,True)].

Extend the code so it will work for triplets.

answer to question 4

```
L = []
for i in (False, True):
    for j in (False, True):
        L.append((i, j))
print L

L = []
for i in (False, True):
    for j in (False, True):
        for k in (False, True):
            L.append((i, j, k))
print L
```

question 5

Write a program to sum the values of coins.

The coins are represented by a sequence of characters, like

`pnqdpqnnppdqddpp`,

where `p`, `n`, `d`, `q` represent respectively a penny (\$0.01), nickel (\$0.05), dime (\$0.10), and quarter (\$0.25). Use a dictionary.

answer to question 5

```
DICTIONARY = {'p':0.01, 'n':0.05, 'd':0.10, 'q':0.25}
DATA = raw_input('give a sequence : ')
S = 0
for C in DATA:
    S = S + DICTIONARY[C]
print 'value = ', S
```

question 6

Give Python code to convert strings like "34 dollar" and "1.23 dollar" into "\$34" and "\$1.23".

The number in front of the `dollar` can be of any size.

answer to question 6

```
INPUT = raw_input('give a string : ')
SPLITTED = INPUT.split(' ')
AMOUNT = float(SPLITTED[0])
if float(int(AMOUNT)) == AMOUNT:
    FORMATTED = '%d' % int(AMOUNT)
else:
    FORMATTED = '%.2f' % AMOUNT
OUTPUT = '$' + FORMATTED
print OUTPUT
```

question 7

Consider the following game.

The computer generates a random natural number between 1 and 9.
This number is the secret.

The user is allowed to make three guesses for the secret.

Write Python code to implement this game.

answer to question 7

```
from random import randint
SECRET = randint(1, 9)
for k in range(3):
    GUESS = raw_input('make a guess : ')
    if SECRET == int(GUESS):
        print 'Congratulations!'
        break
    else:
        print 'Please try again.'
print 'Game over.'
```

question 8

To test whether `random.uniform(0,1)` generates numbers uniformly distributed in $[0, 1]$,

we generate 1000 samples and build a histogram, counting how many numbers are in the four equal subintervals of $[0, 1]$.

Write Python code to perform this test.

answer to question 8

```
from random import uniform as u
L = [u(0, 1) for k in range(1000)]
H = [0 for k in range(4)]
for r in L:
    if r < 0.25:
        H[0] = H[0] + 1
    elif r < 0.50:
        H[1] = H[1] + 1
    elif r < 0.75:
        H[2] = H[2] + 1
    else:
        H[3] = H[3] + 1
print 'histogram :', H
```

question 9

Explain how to simulate an unfair coin that when flipped returns head with probability 0.6.

Give a Python function for such unfair coin.

answer to question 9

```
def unfair_coin():
    "returns head with probability 0.6"
    from random import uniform
    toss = uniform(0, 1)
    if toss < 0.6:
        return 'head'
    else:
        return 'tail'

print 'simulation of an unfair coin'
while True:
    print unfair_coin()
    ANS = raw_input('continue ? (y/n) ')
    if ANS != 'y':
        break
```

question 10

Define a Python function that returns as a tuple the roots of a polynomial $p(x) = ax^2 + bx + c$,

using $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

Write the function so that 1 is the default value for a .

answer to question 10

```
def roots ( cx0, bx1, ax2=1 ):
    "returns roots via quadratic formula"
    from cmath import sqrt
    disc = sqrt(bx1**2 - 4*ax2*cx0)
    return ((-bx1+disc)/(2*ax2), (-bx1-disc)/(2*ax2))

ARAW = raw_input('give a : ')
BRAW = raw_input('give b : ')
CRAW = raw_input('give c : ')
A = float(ARAW)
B = float(BRAW)
C = float(CRAW)
print 'roots = ', roots(C, B, A)
```

question 11

A trigonometric function like $A \sin(k2\pi t)$ has amplitude A and frequency k (number of cycles per time unit).

Write a Python function f to evaluate $A \sin(k2\pi t)$ at t (the first argument of f) where default values for A and k are both one.

If the last optional argument of f takes the value `no_pi`, then the function will return the value $A \sin(kt)$ instead.

answer to question 11

```
def fun(time, amp=1, freq=1, arg='pi'):  
    "return amp*sin(freq*2*pi*time)"  
    from math import pi, sin  
    if arg == 'nopi':  
        return amp*sin(freq*2*time)  
    else:  
        return amp*sin(freq*2*pi*time)
```

```
ARAW = raw_input('give the amplitude : ')  
KRAW = raw_input('give the frequency : ')  
TRAW = raw_input('give the time : ')  
AMP = float(ARAW)  
FRQ = float(KRAW)  
TIM = float(TRAW)  
VAL = fun(TIM, AMP, FRQ)  
print 'the value :', VAL  
VAL = fun(TIM, AMP, FRQ, 'nopi')
```

question 12

Give Python code to create a list of the first 20 natural numbers, raised to the power 3.

Do it once with `for` and once without using any `for` or `while` loops, but do it with list comprehensions.

answer to question 12

```
L = []
for k in range(20):
    L.append(k**3)
print L
L = [k**3 for k in range(20)]
print L
```

question 13

Use list comprehensions to make

```
L = [( 'a' , 97), ( 'b' , 98), ..., ( 'z' , 122)].
```

The list of tuples L links the lower case letters a to z to their respective ASCII codes.

answer to question 13

```
A = [k for k in range(ord('a'), ord('z')+1)]  
B = [chr(k) for k in A]  
L = zip(B, A)  
print L
```

question 14

Write a Python program that doubles the spaces in a file:
every space in the input file is replaced by two consecutive spaces
in the output file.

answer to question 14

```
NAME = raw_input('Give the name of the input file : ')
INFILE = open(NAME, 'r')
NAME = raw_input('Give the name of the output file : ')
OUTFILE = open(NAME, 'w')
while True:
    C = INFILE.read(1)
    if C == '':
        break
    elif C == ' ':
        OUTFILE.write(C)
        OUTFILE.write(C)
    else:
        OUTFILE.write(C)
```