

German University in Cairo
Media Engineering and Technology Faculty
Prof. Dr. Slim Abdennadher
Dr. Aysha El Safty

December 05, 2016

CSEN102: Introduction to Computer Science Winter Semester 2016-2017 Second Chance Midterm Exam

Bar Code

Instructions: Read carefully before proceeding.

- 1) Please tick your major

	Major
	Engineering
	BI

- 2) Duration of the exam: 2 hours (120 minutes).
- 3) No books or other aids are permitted for this test.
- 4) This exam booklet contains 11 pages, including this one. Three extra sheets of scratch paper are attached and have to be kept attached. **Note that if one or more pages are missing, you will lose their points. Thus, you must check that your exam booklet is complete.**
- 5) Write your solutions in the space provided. If you need more space, write on the back of the sheet containing the problem or on the four extra sheets and make an arrow indicating that. **Scratch sheets will not be graded unless an arrow on the problem page indicates that the solution extends to the scratch sheets.**
- 6) When you are told that time is up, stop working on the test.

Good Luck!

Don't write anything below ; -)

Exercise	1	2	3	4	5	6	Σ
Possible Marks	10	14	10	12	16	10	72
Final Marks							

Exercise 1 Conditional Algorithms
Ceiling

(10 Marks)

Write a Python function `ceil` that takes a floating number and returns the next integer after the number x , or x if x is already an integer.

For example,

- `ceil(3.7)` should return 4
- `ceil(2.1)` should return 3
- `ceil(4)` should return 4

Solution:

```
def ceill(x):  
    ↪ x1 = x * 10  
    ↪ d = x1 % 10  
    ↪ if (d == 0):  
        ↪↪ r = x  
    ↪ else:  
        ↪↪ r = int(x) + 1  
    ↪ print(r)  
  
x = eval(input())  
ceill(x)
```

Exercise 2 Conditional Algorithms

(14 Marks)

Write a function `time_to_minutes(s)` that takes a string `s` and returns the number of minutes since midnight. Assume that the string `s` is always of the following form `HH:MM AM` or `HH:MM PM`. For example,

- if `s = '02:45 PM'`, then the function should display 885, which corresponds to $14 * 60 + 45$
- if `s = '09:05 AM'`, then the function should display 545, which corresponds to $9 * 60 + 5$
- if `s = '12:00 AM'`, then the function should display 0

Solution:

```
def time_to_minutes(x):
    r = 0
    if(x[6] == 'A'):
        r = r + (int(x[0])*10 + int(x[1]))*60 + int(x[3])*10 + int(x[4])
    else:
        r = r + (int(x[0])*10 + int(x[1])+12)*60 + int(x[3])*10 + int(x[4])
    print(r)
s = input()
time_to_minutes(s)
```

Exercise 3 Iterative Algorithms

(10 Marks)

Write a Python function `evaluate(p, x)` that returns the evaluated polynomial $p(x)$.

We represent polynomials as a list of floats. In other words `[1.5, -2.2, 3.1, 0, -1.0]` is a representation of $1.5 - 2.2x + 3.1x^2 + 0x^3 - x^4$. We evaluate by substituting in for the value x . For example

- `evaluate([1.5, -2.2, 3.1, 0, -1.0], 2)` is $1.5 - 2.2 \times 2 + 3.1 \times 4 - 1 \times 16 = -6.5$
- `evaluate([2], 4)` is 2

Assume that `p` is a list (`len > 0`) of floats and `x` is a float.

Solution:

```
def evaluate(p, x):
    r = 0
    i = 0
    while(i < len(p)):
        r = r + p[i] * x**i
        i += 1
    print(r)
```

```
p = eval(input())
x = eval(input())
evaluate(p, x)
```

Exercise 4 Iterative Algorithms

(12 Marks)

Write a Python program that takes two lists of integers L1 and L2 as input and creates a list with the element-wise product of L1 and L2. If the lists are of different length, it uses 1 for the missing coefficients.

For example:

- L1 = [2, 2, 3] and L2 = [7, 5, 6], the program should generate and display [14, 10, 18]
- L1 = [2, 2, 3] and L2 = [7, 5, 6, 7], the program should generate and display [14, 10, 18, 7]
- L1 = [2, 2, 3, 8] and L2 = [7, 5, 6], the program should generate and display [14, 10, 18, 8]

Solution:

```
L1 = eval(input())
L2 = eval(input())
i = 0
n1 = len(L1)
n2 = len(L2)

if(n1 >= n2):
    ↪ n = n1
    ↪ m = n2
else:
    ↪ n = n2
    ↪ m = n1

L3 = [1]*n
while(i < m):
    ↪ L3[i] = L1[i] * L2[i]
    ↪ i += 1
while(i < n1):
    ↪ L3[i] = L1[i]
    ↪ i += 1
while(i < n2):
    ↪ L3[i] = L2[i]
    ↪ i += 1
print(L3)
```

Exercise 5 Iterative Algorithms

(8+8=16 Marks)

The made-up language Baablee takes any word and doubles all of the vowels. For example, `bite` becomes `biitee`, `coat` becomes `cooaat`, and `beef` becomes `beeeef`. For the purpose of Baablee, the only vowels are 'a', 'e', 'i', 'o', and 'u'. Implement the functions below which convert a string to its Baableed version and back.

- a) `encode(s)` that takes a string `s` and returns a Baableed copy of `s`. For example, `encode('axe')` is `'aaxee'` and `encode('beef')` is `'beeeef'`. Assume that `s` is a string of only lower-case letters.

Solution:

```
def encode(s):
    n = len(s)
    i = 0
    res = ""
    while i < n:
        if(s[i] == 'a' or s[i] == 'e' or s[i] == 'i' or s[i] == 'o' or s[i] == 'u'):
            res = res + s[i] + s[i]
        else:
            res = res + s[i]
        i += 1
    print(res)

a = input()
encode(a)
```

- b) `decode(s)` that takes a `s` and returns the a copy of `s` restored to original form. For example, `decode('aaxee')` is `'axe'` and `decode('beeeef')` is `'beef'`. Assume that `s` is a Baableed string of only lower case letters.

Solution:

```
def decode(s):
    n = len(s)
    i = 0
    res = ""
    while i < n:
        if(s[i] == 'a' or s[i] == 'e' or s[i] == 'i' or s[i] == 'o' or s[i] == 'u'):
            res = res + s[i]
            i += 1
        else:
            res = res + s[i]
            i += 1
    print(res)

a = input()
decode(a)
```

Exercise 6 Tracing
Mysterious Task

(8+2 = 10 Marks)

```
list_A = eval(input())
m = eval(input())
n = len(list_A)
list_B = [0] * n
i = 0

while (i < m):
    list_B[i] = list_A[n-m+i]
    i += 1

j = 0
i = m
while (i < n):
    list_B[i] = list_A[j]
    j += 1
    i += 1
print(list_B)
```

- a) What does the program display for the following input? Trace the program.

```
list_A = [2, 4, 6, 7, 3, 8]
m = 3
```

Solution:

[7, 3, 8, 2, 4, 6]

- b) What does the program do for any input list and an integer m ?

Solution:

The algorithm rotates the elements in the list m times to the left.

Exercise 7 Iterative Algorithms
Bonus

(Bonus=8 Marks)

Write a Python algorithm that given an integer n consisting of more than one digit, **modifies** the number by swapping each two digits with each other, starting from the right-most digit.

For Example:

- if $n = 2476$, the algorithm should change n to be 4267
- if $n = 214579$, the algorithm should change n to be 125497
- if $n = 235$, the algorithm should change n to be 253

Solution:

```
n = eval(input())
s = 1
p = 0
m = n
res = 0
while (n > 99):
    a = n % 10
    b = (n//10) % 10
    n = n//100
    res = res + (a*(10**s)) + (b*(10**p))
    p +=2
    s +=2

a = n % 10
if( n//10 > 0):
    b = (n//10) % 10
    res = res + (a*(10**s)) + (b*(10**p))
else:
    res = res + (a*(10**(s-1)))
print(res)
```

Scratch paper

Scratch paper

Scratch paper
