Name:

Student ID:

10.009 The Digital World

Term 3. 2016.

Midterm Exam

March 18, 2016 (Friday) 2:30-5:00pm. Room: Cohort Classroom

Most recent update: February 3, 2017

- Write your name and student ID at the top of this page.
- This exam has three parts. First try to complete Part A and Part B, then move on to Part C. You can secure a maximum of 100 points.
- For Part A (questions 1-2), write your answers on this exam paper.
- For Part B and C (questions 3-7), submit your solutions to Tutor. When a question in Part C has a written component, write your answer on this exam paper.
- You may consult all material on your laptop and in your notes. You may also use any book(s) as a reference.
- **You are not allowed to use any Internet accessing or communicating device during the exam.**
- **You are not allowed to consult anyone inside or outside of the classroom other than the proctors in the examination room.**
- Use IDLE / Canopy to test your programs. Once you are satisfied, enter your program into Tutor and either save it or submit. In case you decide to save, then you MUST submit it before the end of the exam.
- All answers will be graded manually. You may be able to earn partial credit for questions.
- Good luck!

Summary:

Category	Description	Number of Problems	Total Points
Part A	Written questions	2 (Questions 1-2)	20
Part B	Short programming questions	4 (Questions 3-5)	25
Part C	Longer or challenging programming questions	2 (Questions 6-7)	55

Q1	
Q2	
Q7b	
SubTotal	

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN
HONOUR CODE

"As a member of the SUTD community, I pledge to always uphold honourable conduct. I will be accountable for my words and actions, and be respectful to those around me."

Introduction to the SUTD Honour Code
What is the SUTD Honour Code?
The SUTD Honour Code was established in conjunction with the school's values and beliefs, in good faith that students are able to discern right from wrong, and to uphold honourable conduct. It is an agreement of trust between the students and the staff and faculty of SUTD, and serves as a moral compass for students to align themselves to. Being in a university that aspires to nurture the leaders of tomorrow, it calls for students to behave honourably, not just solely in their academic endeavours, but also in everyday life.

What the Honour Code encompasses
Integrity & Accountability
To be honourable is to do what is right even when nobody is watching, and to be accountable for the things one does. One should always be accountable for one's words and actions, ensuring that they do not cause harm to others. Putting oneself in a favourable position at the expense of others is a compromise of integrity. We seek to create a community whereby we succeed together, and not at the expense of one another.

Respect
Part of being honourable is also respecting the beliefs and feelings of others, and to never demean or insult them. Should conflicts arise, the aim should always be to settle them in a manner that is non-confrontational, and try to reach a compromise. We will meet people of differing beliefs, backgrounds, opinions, and working styles. Understand that nobody is perfect, learn to accept others for who they are, and learn to appreciate diversity.

Community Responsibility
In addition to that, being honourable also involves showing care and concern for the community. Every individual has a duty to uphold honourable conduct, and to ensure that others in the community do likewise. The actions of others that display immoral or unethical conduct should not be condoned nor ignored. We should encourage each other to behave honourably, so as to build a community where we can trust one another to do what is right.

Student's signature

Part A

Q.1 [10 points]

State the similarities and differences between list and dictionary data structures. State what kind of data is suitable for each data structure and give examples.

Your Answer:

Q.2 [10 points]

A student wrote the following Python program to instruct the robot to move. From the written code, answer the following:

a) Explain briefly what the student tries to do with the code. Give some numerical values in your explanation related to the speed of the robot. [3 pts]

b) How many times is the function increasePower() being called? Explain briefly how you arrived at your answer. [3 pts]

c) Explain what will happen if the sleep(2) statement is removed from the code? [4 pts]

```
from time import sleep
from eBot import eBot

ebot = eBot.eBot()
ebot.connect()

power = 0.1

def increasePower(power):
        power += 0.1
        return power

while power <= 1:
        ebot.wheels(power,power)
        sleep(2)
        power=increasePower(power)
ebot.disconnect()
```

Write your answer on the next page.

Your Answer:

Part B

Q.3 [5 points]

Write a function norm(z1,z2,z3) that returns the Euclidean norm in 3-dimensional complex space C^3, where z1,z2,and z3 are complex numbers. The norm is calculated using the following formula.

$$\|z\| := \sqrt{z_1\overline{z_1} + z_2\overline{z_2} + z_3\overline{z_3}}$$

where $\overline{z_n}$ is the conjugate of z_n and can be computed in Python using z.conjugate().

The function should return a real number rather than a complex number, and the number should be **rounded to three decimal places**. *Hint: You may want to use* cmath.sqrt() *rather than* math.sqrt() *when dealing with complex numbers. You can also obtain the real part of a complex number using its attribute, i.e. z.*real*.*

Sample tests:

```
print 'test 1'
z1=1+3j
z2=-1+3j
z3=-1-3j
ans=norm(z1,z2,z3)
print ans

print 'test 2'
z1=1+2j
z2=-1+2j
z3=-1-2j
ans=norm(z1,z2,z3)
print ans

print 'test 3'
z1=1+1j
z2=-1+1j
z3=-1-1j
ans=norm(z1,z2,z3)
print ans
```

Output:

```
test 1
5.477
test 2
3.873
test 3
2.449
```

Q.4 [10 points]

Write a function factors(n) that takes an integer n as an input and returns a list that includes all the positive number factors of n, where $n \geq 1$. The output returns the list of all the factors in an ascending order. *Hint: you can use the function* sorted() *to sort the output list.*

Sample tests:

```
print 'test 1'
ans=factors(6)
print ans

print 'test 2'
ans=factors(12)
print ans

print 'test 3'
ans=factors(7)
print ans

print 'test 4'
ans=factors(15)
print ans

print 'test 5'
ans=factors(21)
print ans

print 'test 6'
ans=factors(1)
print ans

print 'test 7'
```

Output:

```
test 1
[1, 2, 3, 6]
test 2
[1, 2, 3, 4, 6, 12]
test 3
[1, 7]
test 4
[1, 3, 5, 15]
test 5
[1, 3, 7, 21]
test 6
[1]
test 7
[1, 3, 9]
```

```
ans=factors(9)
print ans
```

Q.5 [10 points]

Write a function combinations(n1,n2) that takes in two integers n1 and n2, and returns a list of tuples of all possible combinations of two integers, both of which range from n1 to n2. The function also returns a second output which is the total number of all combinations. The output combinations must start from the lower number to the higher number as shown in the expression below and in the output test cases. You are not allowed to use any built-in function that gives the same result immediately.

$$output = ([(i,j), i \in (n_1 \dots n_2 - 1), j \in (n_1 + 1 \dots n_2)], total)$$

Sample tests:

```
print 'test 1'
ans=combinations(1,7)
print ans

print 'test 2'
ans=combinations(3,5)
print ans

print 'test 3'
ans=combinations(-1,2)
print ans
```

Output:

```
test1
([(1, 2), (1, 3), (1, 4), (1, 5), (1, 6), (1, 7), (2, 3), (2, 4), (2, 5), (2, 6), (2, 7), (3, 4), (3, 5), (3, 6), (3, 7), (4, 5), (4, 6), (4, 7), (5, 6), (5, 7), (6, 7)], 21)
test 2
([(3, 4), (3, 5), (4, 5)], 3)
test 3
([(-1, 0), (-1, 1), (-1, 2), (0, 1), (0, 2), (1, 2)], 6)
test 4
([], 0)
```

```
print 'test 4'
ans=combinations(0,0)
print ans
```

Part C

Q.6 [Total: 40 points]

Gaussian Elimination is an algorithm for solving systems of linear equations. In this question, you will write a program to perform Gaussian Elimination. **Note that the indices of the rows and columns for the matrices start from 0.** You will need to write and submit four functions:

readMatrix(f)

mulRowByC(matOp,i,c)

addRowMulByC(matOp,i,c,j)

gaussElimination(data)

Note that you need not implement the functions in the sequence shown above. You can start with mulRowByC or addRowMulByC first if you think these two are easier.

Overall Test Cases:

We provide two text files that you can use to test your program, which you can download from **Tutor**. "gauss1.txt" solves a system of linear equations, while "gauss2.txt" uses Gaussian Elimination to find the inverse of a matrix.

Test Code:

```
# replace filename with either 'gauss1.txt' or 'gauss2.txt'
f=open(filename,'r')
data=readMatrix(f)
matA,result=gaussElimination(data)
print matA
print result
```

Output for file "gauss1.txt":

```
[[2.0, 1.0, -1.0, 8.0], [-3.0, -1.0, 2.0, -11.0], [-2.0, 1.0, 2.0, -3.0]]
[[1.0, 0.0, 0.0, 2.0], [0.0, 1.0, 0.0, 3.0], [-0.0, -0.0, 1.0, -1.0]]
```

Output for file "gauss2.txt":

```
[[2.0, -1.0, 0.0, 1.0, 0.0, 0.0], [-1.0, 2.0, -1.0, 0.0, 1.0, 0.0], [0.0, -1.0, 2.0, 0.0, 0.0, 1.0]]
[[1.0, 0.0, -0.0, 0.75, 0.5, 0.25], [0.0, 1.0, -0.0, 0.5, 1.0, 0.5], [0.0, 0.0, 1.0, 0.25, 0.5, 0.75]]
```

(a) [10 points]

readMatrix(f):

This function takes a file object f as input (note: f is NOT the name of a file, but a file object). The file referred to by f contains several lines of data (see the test case below). The file consists of two parts. The first part is a DATA section which contains the elements of a matrix to be solved using Gaussian Elimination. The second part is the OP section which contains the list of operations to be performed when doing Gaussian Elimination.

Example file 'gauss2.txt':

```
DATA
2 -1 0 1 0 0
-1 2 -1 0 1 0
0 -1 2 0 0 1
OP
2 0 0.5 1
1 1 0.666666666667
2 1 1 2
1 2 0.75
2 2 0.666666666667 1
2 1 1 0
1 0 0.5
END
```

The function readMatrix(f) reads the matrix data and its operations from the file and returns a dictionary. The dictionary contains two keys. The first key is 'matrix' and the second one is 'op'. The 'matrix' key contains a matrix represented as a list of lists. Similarly, the 'op' key contains the list of operations as a list of list where each element in the list is a string.

For the above example, the dictionary returned by readMatrix(f) will be:

output: {'matrix': [[2.0, -1.0, 0.0, 1.0, 0.0, 0.0], [-1.0, 2.0, -1.0, 0.0, 1.0, 0.0], [0.0, -1.0, 2.0, 0.0, 0.0, 1.0]], 'op': [['2', '0', '0.5', '1'], ['1', '1', '0.666666666667'], ['2', '1', '1', '2'], ['1', '2', '0.75'], ['2', '2', '0.666666666667', '1'], ['2', '1', '1', '0'], ['1', '0', '0.5']]}

Note: the dictionary will be unordered. While grading, your returned dictionary will be automatically ordered before comparing it with the Tutor output.

(b) [10 points]

mulRowByC(matOp,i,c):

This function takes as input a list of lists representing a matrix, an index i, and a constant c. The argument matOp is the matrix on which the operation is to be performed. The argument i is the index of the row which is to be multiplied by the constant. The argument c is the constant by which the row is to be multiplied. **The function should return a new matrix as a result of doing the following operation**:

$$row_i \times c \rightarrow row_i$$

For example,

$matA = \begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 1 & 0 \\ 0 & -1 & 2 & 0 & 0 & 1 \end{bmatrix}$ and i = 2, c = 3

should result in

$$result = \begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 1 & 0 \\ 0 & -3 & 6 & 0 & 0 & 3 \end{bmatrix}$$

Test Case:

```
A=[[0,2,1,-1],[0,0,3,1],[0,0,0,0]]

ans=mulRowByC(A,0,2)

print ans
```

Output:

```
[[0, 4, 2, -2], [0, 0, 3, 1], [0, 0, 0, 0]]
```

(c) [10 points]

addRowMulByC(matOp,i,c,j):

This function takes as input a list of lists representing a matrix, an index i, a constant c, and another index j. The argument matOp is the matrix on which the operation is to be performed. The argument i is the index of the row index which is to be multiplied by the constant c. The argument c is the constant by which the row i is to be multiplied. The argument j is the index of the row in matOp to which the product of row i and constant c is to be added. **The function should return a new matrix as a result of doing the following operation**:

$$row_i \times c + row_j \rightarrow row_j$$

For example,

$matA = \begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 1 & 0 \\ 0 & -1 & 2 & 0 & 0 & 1 \end{bmatrix}$ and i = 2, c = 3, j = 0

should result in

$$result = \begin{bmatrix} 2 & -4 & 6 & 1 & 0 & 3 \\ -1 & 2 & -1 & 0 & 1 & 0 \\ 0 & -1 & 2 & 0 & 0 & 1 \end{bmatrix}$$

Test Case:

```
A=[[0,2,1,-1],[0,0,3,1],[0,0,0,0]]

ans=addRowMulByC(A,0,0.5,1)

print ans
```

Output:

```
[[0, 2, 1, -1], [0.0, 1.0, 3.5, 0.5], [0, 0, 0, 0]]
```

(d) [10 points]

gaussElimination(data):

This function takes as input a dictionary data created by readMatrix(f) and returns a tuple. The function performs the list of operations from the first item in the list to the last one in sequence and returns the original matrix and the final matrix after all the operations have been performed. This function should call the next two functions listed below to perform the matrix operations as follows:

If the first element of the operation sublist is '1', the operation “Multiply Row by Constant C” should be performed by calling the function mulRowByC(matOp,i,c). The argument matOp is the matrix on which the operation is to be performed. It is either the original matrix or the result of the previous operation.

The argument i is the index of the row which is to be multiplied by the constant; it is taken from the second element of the operation sublist. The argument c is the constant by which the row is to be multiplied, and it is taken from the third element of the operation sublist.

For example,

input: ['1', '2', '0.75']

should call mulRowByC(matOp, 2, 0.75).

If the first element of the operation sublist is '2', the operation "Add Row by Constant C" should be performed by calling the function addRowMulByC(matOp,i,c,j). The argument matOp is the matrix on which the operation is to be performed. It is either the original matrix or the result of the previous operation. The argument i is the index of the row to be multiplied by the constant c. The argument c is the constant by which row i in matOp is to be multiplied. The argument j is the row in matOp which the product of row i and constant c is to be added.

For example,

input: ['2', '1', '1', '2']

should call addRowMulByC(matOp,1, 1, 2).

The function should return the original matrix and the final matrix where **each element is rounded to two decimal places**. **Note that you should round the output only after the last operation and not at the end of each operation.**

Test Case:

```
data = {'matrix': [[2.0, -1.0, 0.0, 1.0, 0.0, 0.0], [-1.0, 2.0, -1.0, 0.0, 1.0, 0.0], [0.0, -1.0, 2.0, 0.0, 0.0, 1.0]], 'op': [['2', '0', '0.5', '1'], ['1', '1', '0.666666666667'], ['2', '1', '1', '2'], ['1', '2', '0.75'], ['2', '2', '0.666666666667', '1'], ['2', '1', '1', '0'], ['1', '0', '0.5']]}

matA, result=gaussElimination(data)

print matA

print result
```

Output:

```
[[2.0, -1.0, 0.0, 1.0, 0.0, 0.0], [-1.0, 2.0, -1.0, 0.0, 1.0, 0.0], [0.0, -1.0, 2.0, 0.0, 0.0, 1.0]]

[[1.0, 0.0, -0.0, 0.75, 0.5, 0.25], [0.0, 1.0, -0.0, 0.5, 1.0, 0.5], [0.0, 0.0, 1.0, 0.25, 0.5, 0.75]]
```

Q.7 [15 points] **This question comprises of two parts: (a) Programming, and (b) Written.**

Write a function maxProductThree(num) that returns the maximum product from any three numbers in a list of integers. The list may contain both positive, zero, negative integers, and duplicates, but the maximum product can only be either a negative or a positive number. You can assume that the list contains at least three non-zero numbers so the maximum product will never be zero.

Test Case:

```
num=[6,-3,-10,0,2]

print maxProductThree(num)
```

Output:

```
180
```

a) (10 pts)

Write and submit your function definition in Tutor. You may want to do part (b) before submitting your answer to Tutor.

b) (5 pts, you only get these points if you get Part (a) correctly)

If you write the solution without brute-force method, i.e. finding all possible combinations, there are only a few possible groups of cases. State all the possible groups of cases. For each case, give a sample input with its corresponding output. Write your answer below.

Your Answer:

End of Exam Paper

[this page is left blank]

[this page is left blank]