NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 1 EXAMINATION 2013-2014

CE1003/CZ1003 – INTRODUCTION TO COMPUTATIONAL THINKING

Nov/Dec 2013 Time Allowed: 2 hours

INSTRUCTIONS

1. This paper contains 4 questions and comprises 7 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.

---

1. (a) Why are random numbers generated by computer programs generally pseudorandom?

(3 marks)

(b) Write a Python program that reads a floating point number x, and a positive integer n from the user, and then computes and displays the value of sin(x) using the following series up to (n+1) terms:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \cdots = \sum_{k=0}^{n} \frac{(-1)^k x^{2k+1}}{(2k+1)!}$$

No error checking is needed on the user input.

(10 marks)

(c) Why does the Python program shown below print out `1.0` rather than `1.0000000000000001`?

```
a = 1
b = 0.0000000000000001
print( a + b )
```

(3 marks)

Note: Question No. 1 continues on Page 2

(d) In this question, you have to complete a Python program to print out big letters of NTU with each letter occupying a space of `n` by `n` characters, and one empty column in-between them. For example, if `n` is 5, the big letters to be printed out are:

```
#    # ##### #    #
##   #   #   #    #
# #  #   #   #    #
#  # #   #   #    #
#   ##   #   #    #
#    #   #   #####
```

Carefully study the code below and complete the missing conditions:

```
n = int( input( "input n: " ) )

for y in range(n):
    for x in range(n):
        if (      ): # your code for condition 1
            print("#",end="")
        else:
            print(" ",end="")
    print(" ",end="") # skip
    for x in range(n):
        if (      ): # your code for condition 2
            print("#",end="")
        else:
            print(" ",end="")
    for x in range(n+1):
        if (      ): # your code for condition 3
            print("#",end="")
        else:
            print(" ",end="")
    print()
```

On the answer book, you do not need to write the entire program, but just your code for the three missing conditions. You can also assume that the input `n` is always an odd number between 2 and 20.

(9 marks)

2. (a) (i) Explain what the following code does.

```
a = int( input( "input a:" ) )
b = int( input( "input b:" ) )
if a % b == 0 and a > b > 0:
    print( "Yes. Condition is true." )
```

(3 marks)

(ii) Running the code above may result in an error. Identify and explain this error. Then, describe how to fix it.

(4 marks)

Note: Question No. 2 continues on Page 3

(b) Trace the following program *carefully* and write down what it *prints out*.

```
a = b = c = d = e = 0
for i in range(2,-2,-2):
    a = a + 1
print(a)

for i in range(2):
    b += b+1
print(b)

while c <= 3:
    if 0 == c // 2:
        c = c + 2
    else:
        c -= 1
    c += 2
print(c)

while d <= 5:
    d = d + 2
    if 6 % d == 1:
        break
    d = d + 1
print(d)

while e >= 0:
    if e % 10 == 3:
        break
    e = e*e - e*2 + 4
    if e > 50:
        e = e % 50
print(e)
```

(10 marks)

(c) Write a Python program that keeps asking users for positive float values as its inputs until seeing a sentinel value of -1. The program should then print out the largest two values among all the inputs. No error checking is needed on the inputs and there are at least two inputs. You cannot use list and dictionary in your program.

(8 marks)

3. (a) Give the output of the following code:

```
myStr = "ComputationalThinking"
code = "1003"
print( myStr[2] )
print( myStr[-2] )
print( myStr[:6] + myStr[-3:] )
print( myStr[13:-3] * len(code) )
print( code[::-1] )
newStr = ""
for i in range(0, len(myStr)):
      if myStr[i] in myStr[(i+1):]:
            newStr += myStr[i]
print(newStr)
```

(7 marks)

(b) Create the following lists using list comprehension.

(i) A list of all the odd numbers between 0 and 100.

(3 marks)

(ii) A list of all the digits ('0' to '9') in a string `str`.

(3 marks)

(c) The following Python program asks the user to input an integer and then prints it on the screen. Complete the program by filling in the boxes **A** to **C** each with a single line of code such that the program keeps prompting the user until a valid integer is entered.

```
number_str = input("Input an integer: ")
while True:
      try:
            [ A ]
            [ B ]
      except ValueError:
            [ C ]
print(number_int)
```

(6 marks)

Note: Question No. 3 continues on Page 5

(d) Give the output of the following program:

```
def myFun(list1, list2, number):
      list1.append(number)
      list1 = [3, 6, 9]
      number += 1
      list2[0] = number
      list2.append(list1[0])

list1 = [1, 2, 3]
list2 = [2, 4, 6]
number = 7
for i in range(5):
      myFun(list1, list2, number)
print(list1)
print(list2)
```

(6 marks)

4. (a) Given a list `myList` of strings, write a Python function to create a new list `newList` of strings such that the first string in `newList` concatenates the first and last strings in `myList`, the second string in `newList` concatenates the second and second-to-last strings in `myList`, and so on. If the length of `myList` is odd, the central string in `myList` is appended to the end of `newList`. For example, if `myList` = ["ab", "cd", "ef", "gh", "ij", "kl"], then `newList` = ["abkl", "cdij", "efgh"]; and if `myList` = ["ab", "cd", "ef", "gh", "ij"], then `newList` = ["abij", "cdgh", "ef"]. A skeleton of the function is shown below. Complete the function by filling in the boxes **A** to **E** each with a single line of code to implement the above functionalities.

```
def foldList(myList):
      newList = []
      i = [ A ]
      j = [ B ]
      while i < j:
            [ C ]
            i += 1
            j -= 1
      if [ D ]:
            [ E ]
      return newList
```

(5 marks)

Note: Question No. 4 continues on Page 6

(b) Design a Python program that reads two plain text files (`in1.txt` and `in2.txt`) and produces a new file (`out.txt`) that interleaves the lines of the input files. For example, if the contents of file `in1.txt` are

```
in1 line1
in1 line2
in1 line3
```

and the contents of file `in2.txt` are

```
in2 line1
in2 line2
in2 line3
in2 line4
in2 line5
```

then the output file `out.txt` should have the following contents:

```
in1 line1
in2 line1
in1 line2
in2 line2
in1 line3
in2 line3
in2 line4
in2 line5
```

Note that if the two input files do not have the same number of lines, the remaining lines of the longer file should be appended to the output file. A skeleton of the program is shown below. Complete the program by filling in the boxes **A** to **I** each with a single line of code to implement the above functionalities.

```
inFile1 = [ A ]
inFile2 = [ B ]
lines1  = [ C ]
lines2  = [ D ]
outFile = [ E ]
for i in [ F ]:
    outFile.write(lines1[i])
    outFile.write(lines2[i])
if [ G ]:
    [ H ]
else:
    [ I ]
inFile1.close()
inFile2.close()
outFile.close()
```

(9 marks)

Note: Question No. 4 continues on Page 7

(c) Consider a dictionary that maps employees to their ages. Suppose that you will repeatedly look up the employees based on their ages. Then, it may be helpful to build an inverted index that shows all the employees of each age. For example, if the dictionary is:

```
{"Alice": 26, "Bob": 35, "David": 35,
 "Eric": 42, "Fred": 35, "Grace": 26}
```

the inverted index would be:

```
26: "Alice", "Grace"
35: "Bob", "David", "Fred"
42: "Eric"
```

(i) What would be an appropriate data structure to store an inverted index?

(2 marks)

(ii) Write a Python function `invertedIndex(myDict)` that constructs and returns the inverted index given a dictionary `myDict`.

(9 marks)

END OF PAPER

CE1003 INTRODUCTION TO COMPUTATIONAL THINKING
CZ1003 INTRODUCTION TO COMPUTATIONAL THINKING

Please read the following instructions carefully:

1. **Please do not turn over the question paper until you are told to do so. Disciplinary action may be taken against you if you do so.**

2. You are not allowed to leave the examination hall unless accompanied by an invigilator. You may raise your hand if you need to communicate with the invigilator.

3. Please write your Matriculation Number on the front of the answer book.

4. Please indicate clearly in the answer book (at the appropriate place) if you are continuing the answer to a question elsewhere in the book.