

NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER 2 EXAMINATION 2015-2016

CE1003/CZ1003 – INTRODUCTION TO COMPUTATIONAL THINKING

Apr/May 2016

Time Allowed: 2 hours

INSTRUCTIONS

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

-
1. (a) What are the differences between an algorithm and a program?
(2 marks)
 - (b) What is a machine language? Briefly elaborate on its properties.
(3 marks)
 - (c) An algorithm is needed to search for prime numbers. A prime number is a positive integer greater than 1 that has no positive divisors other than 1 and itself.
 - (i) Use a flowchart to describe an algorithm that prints out prime numbers starting from 1 to k (the upper bound of the search for prime numbers).
(8 marks)
 - (ii) Write a pseudo-code of the algorithm described in Q1(c)(i). Your pseudo-code should read the upper bound, k , from the user.
(7 marks)

Note: Question No. 1 continues on Page 2

- (iii) A Mersenne prime is a prime number that is one less than a power of two. They can be written in the form of $M^n = 2^n - 1$ for some integer n . Modify your pseudo code in Q1(c)(ii) to indicate if the prime number found is also a Mersenne prime. (5 marks)
2. (a) (i) What is a namespace? Briefly elaborate on the Local, Enclosed, Global, Built-in (LEGB) rule. (3 marks)
- (ii) What is duck-typing? How is it used by Python? (2 marks)
- (b) (i) Float is an approximation to the real number. Explain with an example how the limited precision of float results in inexact computation. (5 marks)
- (ii) Describe a method of comparing two float numbers to determine if they are equivalent. Elaborate on the design consideration and limitations of your method. Write this in Python code. (6 marks)
- (iii) Given a constraint of using only integers, describe a method of approximating the division of 2 numbers up to 3 decimal places. Describe the limitations of your method. Write this in a pseudo code. (4 marks)
- (c) In the following Python code segment, where we define a loop to count up to 42, there are a number of syntactical errors. Describe the errors and suggest fixes.
- ```
spam == 0
print('I start with ' + spam + ' number of spam(s).)

while (spam < 42)
spam++
print('Spam exceeds 41')

// Program ends
```
- (5 marks)

3. (a) An integer number is *palindromic* if it is positive and has digits that read the same backwards and forwards. For example, 3773 is palindromic. Using `string`, write a Python program to check if an integer `N` is palindromic. Your program must obtain `N` as input from the user, and print out `True` if the integer is palindromic, and `False` otherwise.

(8 marks)

- (b) Describe two differences between the `string` and `set` data structures.

(4 marks)

- (c) Write a Python function that takes an eight-digit phone number (stored as a string with a hyphen separator in the center, e.g., '1053-4000') as input, and returns the number of unique digits. For example, the phone number '1053-4000' has 5 unique digits, while '9999-1211' has only 3.

(8 marks)

- (d) Determine the printed output of the following program, and briefly explain the functionality of the program.

```
C = {0.5:'Fifty', 0.2:'Twenty', 0.1:'Ten', \
 0.05:'Five', 1.0:'Dollar'}
N = 3.25
while N > 0:
 if N >= 1.0:
 N = N - 1.0
 print(C[1.0])
 elif N >= 0.5:
 N = N - 0.5
 print(C[0.5])
 elif N >= 0.2:
 N = N - 0.2
 print(C[0.2])
 elif N >= 0.1:
 N = N - 0.1
 print(C[0.1])
 else:
 N = N - 0.05
 print(C[0.05])
```

(5 marks)

4. (a) What is the output of the following Python program? Explain your answer. For clarity, indicate a space character with “\_” (underscore) in your answer(s) if necessary.

```
str = "A python is a constricting snake"
print(str[::-1][-3:-9:-1])
```

(4 marks)

- (b) What happens to the contents of an existing file when it is opened for writing by using `open('filename','w')`? What about for appending by using `open('filename','a')`?

(4 marks)

- (c) What is a user-defined module? What are the differences between a function and a user-defined module?

(3 marks)

- (d) Describe the two common approaches to handle exceptions in Python. Discuss their differences.

(6 marks)

- (e) What is the printed output of the script in Figure Q4, if the inputs are `X=10`, `Y=0`, and `Z=4`?

```
def div(A,B):
 return A//B

X=int(input('X='))
Y=int(input('Y='))
Z=int(input('Z='))
try:
 print('X div Z is ',div(X,Z))
 print('Y div X is ',div(Y,X))
 print('X div Y is ',div(X,Y))
 print('Z div X is ',div(Z,X))
except:
 print('Something failed!')

print('All done')
```

**Figure Q4**

(8 marks)

END OF PAPER

**NANYANG TECHNOLOGICAL UNIVERSITY**

**SEMESTER 1 EXAMINATION 2015-2016**

**CE1003/CZ1003 – INTRODUCTION TO COMPUTATIONAL THINKING**

Nov/Dec 2015

Time Allowed: 2 hours

**INSTRUCTIONS**

1. This paper contains 4 questions and comprises 6 pages.
  2. Answer **ALL** questions.
  3. This is a closed-book examination.
  4. All questions carry equal marks.
- 

1. (a) In mathematics,  $\tan \theta = \sin \theta / \cos \theta$ , but the Python code below behaves strangely.

```
import math
a = float(input("input angle: "))
if math.tan(a) == math.sin(a)/math.cos(a):
 print("The formula is correct!")
else:
 print("The formula is incorrect!")
```

Identify and describe the problem in the code, and then suggest how we can fix it. You can assume that there is no error in user input.

(5 marks)

- (b) Fibonacci sequence (1, 1, 2, 3, 5, 8, 13, ...) is an integer sequence, where

$$\begin{aligned} F_1 &= 1 \\ F_2 &= 1 \\ \text{and } F_i &= F_{i-1} + F_{i-2} \quad \text{if } i > 2. \end{aligned}$$

One interesting property about Fibonacci sequence is that the ratio between two successive integers ( $F_{i+1}/F_i$ ) tends to be the famous golden ratio with increase in  $i$ . Write a Python program using multiple assignment to print out the first fifty ratios, i.e.,  $F_2/F_1$ ,  $F_3/F_2$ ,  $F_4/F_3$ , etc.

(6 marks)

Note: Question No. 1 continues on Page 2

- (c) Trace the following program carefully and write down what it prints out.

```
import random
import math

a = b = c = d = e = 1

for i in range(1,-1,1):
 a += i
print(a)

for i in range(1000):
 b = -1 - b
print(b)

while c >= 0:
 c = c + 1
 if c // 3 == 0 :
 continue
 c = 2 - c
print(c)

while d % 2 != 0 or d % 3 != 0:
 d = d + 7
else:
 d = d / 6
print(d)

while e < 3:
 e += random.randint(1, 7)
 e = int(math.sqrt(e))
print(e)
```

(10 marks)

- (d) Explain the meaning and purpose of the symbol \ in Python.

(4 marks)

2. (a) A happy number is defined by the following process: Given a positive integer (say  $n$ ), we compute the sum of squares of its digits. If the sum is 1,  $n$  is said to be a happy number. If the sum is 4,  $n$  is said to be a sad number. Otherwise, we repeat the above process with the resulting sum, until the sum becomes 1 or 4. For example, if we start with 91,

$$9^2 + 1^2 = 82$$

$$8^2 + 2^2 = 68$$

$$6^2 + 8^2 = 100$$

$$1^2 + 0^2 + 0^2 = 1.$$

Hence, 91 is a happy number.

Note: Question No. 2 continues on Page 3

Write a Python program to read a positive integer from the user, and then compute and print out if the input integer is a happy number or not. You can assume that there is no error in user input.

(12 marks)

- (b) The program below should print out a hollow square of size N by N without corners, together with a big cross inside the square.

```
N = int(input("input N: "))
for y in range(1,N+1):
 for x in range(1,N+1):
 if (y == 1 or y == N) and 0 < x < N-1:
 print("#",end="")
 elif (x == 1 or x == N) and 1 < y < N:
 print("#",end="")
 elif 1 < x < N and x == y or x == N-y:
 print("#",end="")
 else
 print(" ",end="")
 print()
```

The program above should print out the patterns below:

```
input N: 8
#####
##
#
#
#
#
##
#####
```

```
input N: 5
###
##
#
##
###
```

However, the program has a number of errors. Read the program code above carefully, identify the errors, and describe how to fix each of them. You may assume that N is an integer and  $N > 4$ .

(10 marks)

- (c) Is there any difference between the following two pieces of Python code?

```
Code #1: if y == 1/x and x+1 != 1:
Code #2: if x+1 != 1 and y == 1/x:
```

If yes, explain the difference between them. If no, explain a potential error in the code.

(3 marks)

3. (a) (i) Write a Python statement to create a string variable *a* such that the statement *print (a)* will output the following message:

C:\Python3.4\test

(2 marks)

- (ii) Given the string *b* = “*I am a student of NTU*”, what is the value of *b[::-1]*?

(2 marks)

- (iii) Suppose *c* = “*xyxyxyxy*”. Write an expression to create another string *d* = “*xxxx*” by removing all the ‘*y*’s from *c*.

(2 marks)

- (iv) Define a function that takes two strings as arguments. The function will compare the two strings and return the smaller string. If the two strings are the same, the function will return any one of them.

(4 marks)

- (b) (i) Given *e* = [1, 4, 3, 2] and *f* = *e.append(0)*, what is the value of *f*? Explain your answer.

(3 marks)

- (ii) Define a function that takes a tuple of numbers as an argument. The function will sort the numbers in the tuple and return the sorted tuple.

(4 marks)

- (c) (i) Design a data structure using composite types to store the student names and selected courses. Here we assume the student names are unique. Each student has a list of selected courses and each course has a course code and course subject. Then define a function that takes a student name as an argument. The function will print out the student name and the courses that the student selected. The format of the output is:

```
Student Name: xxxx
Course Code Course Subject
CZ0000 YYYYYY
CZ9999 zzzzzz
...
```

(6 marks)

Note: Question No. 3 continues on Page 5



- (ii) Suppose a student whose name is Tom has selected the course CZ1003 Introduction to Computational Thinking and the course CZ1001 Discrete Mathematics. Write a Python program that creates a variable using the designed data structure to store the student name and the selected courses, and then calls the function defined in Q3(c)(i) to print out Tom's course list.

(2 marks)

4. (a) Write a recursive function that takes a non-negative number  $n$  as the input and returns the value of  $2^n$ .

$$2^n = \begin{cases} 2^{n-1} \cdot 2 & \text{if } n > 1 \\ 1 & \text{if } n = 0 \end{cases}$$

(5 marks)

- (b) The raw data of a game play has been recorded in a text-based data file. The first line contains the player's name and the starting time separated by ','. Starting from the second line, every 11 lines record the data of play for one attempt and the last line of each attempt is the score of the play in this attempt. The play can be interrupted anytime so the last attempt of play may not have the full record in the file.

- (i) Write a function that takes a file name as an argument and returns the player name and the highest score from all the attempts. Exception handler is required to check if the file exists. If the file cannot be found, the error message "File does not exist." will be printed out. Otherwise, the message "Data has been retrieved." will be printed out before the function returns the result.

(8 marks)

- (ii) Write a Python code to call the function defined in Q4(b)(i) to get the player's name and highest score using the data file "gameData.dat", and then append the data into the file "gameScore.dat".

(5 marks)

Note: Question No. 4 continues on Page 6

- (c) The following code prints “*red*” and “*green*” alternatively when it is called. But it has an error.

```
def func():
 if flag == 1:
 print("red")
 flag = 0
 else:
 print("green")
 flag = 1
flag = 1
func()
func()
```

- (i) Identify the error and explain your answer. (4 marks)
- (ii) Fix the error and write the whole corrected program. (3 marks)

**NANYANG TECHNOLOGICAL UNIVERSITY**

**SEMESTER 2 EXAMINATION 2014-2015**

**CE1003/CZ1003 – INTRODUCTION TO COMPUTATIONAL THINKING**

Apr/May 2015

Time Allowed: 2 hours

**INSTRUCTIONS**

1. This paper contains 4 questions and comprises 5 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.

---

1 (a) (i) What is the common functionality existed in both compilers and interpreters?  
(2 marks)

(ii) Name one advantage of interpreter over compiler and elaborate how an interpreter supports such an advantage.  
(3 marks)

(b) The Python code below aims to constantly regulate the temperature of a freezer, so that the temperature stays in between -3 and -5 degrees.

```
while True
 wait() # wait for one minute
 temp = getTemp() # get current temperature (in float)
 if temp < -3.0:
 raiseTemp() # raise (increase) temperature
 else:
 lowerTemp() # lower (decrease) temperature
 tempInt = int(temp + 0.5) # rounding
 showTemp(tempInt) # display temperature (in integer)
```

However, there are a number of errors in the code. Please identify the errors and then describe how to fix each of them.

(8 marks)

Note: Question No. 1 continues on Page 2

- (c) Write a Python program that reads an integer from the user, which is the width of the pattern below, and then prints out the pattern. You may assume that the input integer is at least 3. No error checking on the user input is required. Hint: use nested for loops and `print("x",end="")`.

Here are the sample runs:

```
input the width of the pattern: 3
x
xx
xxx
x x
xxx
xx
x
```

```
input the width of the pattern: 5
x
xx
xxx
xxxx
xxxxx
x x
x x
x x
xxxxx
xxxx
xxx
xx
x
```

(12 marks)

- 2 (a) (i) What are tokens during the processing of interpreters? (2 marks)

- (ii) There are different types of tokens. Name one token type and describe its usage. (2 marks)

- (b) Write a Python program that reads a positive integer  $n$  from the user, and displays the sum of the last five digits of the factorial of  $n$ .

For example, if  $n$  is 10, the factorial of  $n$  is 3628800, so the last five digits is 28800 and their sum is 18. No error checking on the user input is required, and  $n$  is at least 10.

(11 marks)

Note: Question No. 2 continues on Page 3

- (c) Trace the following program carefully and write down what it prints out.

```
import random

a = b = c = d = e = 0

for i in range(100):
 a = 1 - a
print(a)

for i in range(1,3,5):
 b += i
print(b)

b = 0
while c < 3:
 c,b = b,c+1
 if c // 2 == 0 :
 continue
 c = c + 2
print(c)

while d % 2 != 1 or d % 3 != 1:
 d = d + 5
else:
 d = d / 5
print(d)

while e != 7:
 e = e + random.randint(1, 5)
 if e > 50:
 e = 0
print(e)
```

(10 marks)

3. (a) Write a Python statement to define a string variable with the following sentence.

Hart said: "It is, but hadn't ought to be."

(3 marks)

- (b) Describe the rule that unqualified namespace follows.

(5 marks)

Note: Question No. 3 continues on Page 4

(c) (i) When will you need to use collective data structures? Give two examples. (3 marks)

(ii) Describe two differences between the string and set data structures. (4 marks)

(iii) Define what recursion is and why it is different from looping. (4 marks)

(d) (i) Trace the following program carefully and write down what the program prints out.

```
myText = "123456"
```

```
for i in range(2, 10):
 print(myText[i])
```

(4 marks)

(ii) Re-write the program in Q3d(i) such that it prints out the contents of `myText` in reversed order without using the `for/while` loop. (2 marks)

4. (a) (i) Why should you specify the file encoding when opening a file? (2 marks)

(ii) Write a Python statement to open the file `try/test.txt` with an encoding `utf-8`. (4 marks)

(b) Describe the main components of defining a procedure in Python. (6 marks)

(c) What is an exception? Give an example. (2 marks)

Note: Question No. 4 continues on Page 5

- (d) Describe the two common approaches to handle exceptions in your code. Discuss their differences.  
(6 marks)
  
- (e) Write a Python program that reads a file called `test.txt`. Your program has to consider at least one exception.  
(5 marks)

END OF PAPER

**NANYANG TECHNOLOGICAL UNIVERSITY****SEMESTER 1 EXAMINATION 2014-2015****CE1003/CZ1003 – INTRODUCTION TO COMPUTATIONAL THINKING**

Nov/Dec 2014

Time Allowed: 2 hours

**INSTRUCTIONS**

1. This paper contains 4 questions and comprises 5 pages.
2. Answer **ALL** questions.
3. This is a closed-book examination.
4. All questions carry equal marks.

- 
- 1 (a) Explain why the interpreter approach allows cross-platform program execution while the compiler approach does not. (4 marks)

- (b) Write a Python program that reads the number of terms from the user, and then displays the result of the summation series below:

$$\frac{1}{1} + \frac{1}{3} + \frac{1}{6} + \frac{1}{10} + \frac{1}{15} + \frac{1}{21} + \dots$$

Note that the denominator of the 2nd term can be computed by adding 2 to the denominator of the 1st term; the denominator of the 3rd term can be computed by adding 3 to the denominator of the 2nd term, etc. No error checking on the user input is required.

(8 marks)

- (c) In Python, is it always possible to rewrite a `while` loop as a `for` loop? If no, explain why. If yes, describe how.

(3 marks)

Note: Question No. 1 continues on Page 2



- (d) Trace the following program carefully and write down what it prints out.

```

a = b = c = d = e = 0

for i in range(3):
 a = a + 1
print(a)

for i in range(4,0,-4):
 b += i
print(b)

while c < 4:
 if c % 3 == 1:
 break
 c = c + 2
else:
 c = c + 1
print(c)

while d < 15:
 if d <= 4:
 d = 10 / (d + 1)
 d = d + 3
print(d)

while e < 12:
 e = e + 1
 if e > 10:
 e = 100 // e
 continue
 e = e + 1
print(e)

```

(10 marks)

2. (a) (i) Given two integer variables (a=10 and b=3) in a Python program, someone develops the following code to swap their data values:

```

a = a - b # statement 1
b = b - a # statement 2
a = b - a # statement 3

```

However, there is an error in the code. Describe how you fix the error. Note that you can only modify one of the three statements.

(3 marks)

- (ii) In Python, can we swap the values of two integer variables by one instead of three statements? If so, how would you do it?

(2 marks)

Note: Question No. 2 continues on Page 3

- (4 marks)

- (8 marks)

- ```
Nlevels = int( input("How many levels: ") )
n       = Nlevels
for level in range( 1 , Nlevels*2+2 ):
    # odd rows on top and main branch on bottom
    if ( level % 2 == 0 ):
        for i in range( Nlevels ):
            print( " " , end="" )
        print("|")
    # even rows on top
else:
    for i in range( n ):
        print( " " , end="" )
    for i in range( Nlevels-n ):
        print( "-" , end="" )
    print( "+" , end="" )
    for i in range( Nlevels-n ):
        print( "-" , end="" )
    n = n - 1
print()
```

```

How many levels: 2
|
--+-
|
|
|
|

How many levels: 4
|
--+-
|
--+-
|
-----+-
|
-----+-
|
|
|
|
|

```

However, the program has *several* errors. Identify the errors and describe how to fix each of them.

3

3. (a) Using strings, write a Python program to check if an integer N is palindromic, which means N is positive and has digits that read the same backwards and forwards. For example, 3773 is palindromic. Your program must obtain N as input from the user, and print out `True` if the integer is palindromic, and `False` otherwise.

(6 marks)

- (b) Describe briefly the following Python composite data types:

(i) Set

(3 marks)

(ii) Dictionary

(3 marks)

- (c) Write a Python function that takes as input an eight-digit phone number, stored as a string with a hyphen separator, and returns `True` if the phone number has at least 4 unique digits, and `False` otherwise. For example, the phone number '1053-4000' has 5 unique digits, while '9999-1211' has only 3.

(8 marks)

- (d) Determine the printed output of the following program, and explain your reasoning.

```
C = {0.5:'Fifty', 0.2:'Twenty', 0.1:'Ten', \
     0.05:'Five', 1.0:'Dollar'}
N = 2.45
while N>0:
    if N>=1.0:
        N=N-1.0
        print(C[1.0])
    elif N>=0.5:
        N=N-0.5
        print(C[0.5])
    elif N>=0.2:
        N=N-0.2
        print(C[0.2])
    elif N>=0.1:
        N=N-0.1
        print(C[0.1])
    else:
        N=N-0.05
        print(C[0.05])
```

(5 marks)

4. (a) What happens to the contents of an existing file when it is opened for writing by using `open('filename','w')`? What about for appending by using `open('filename','a')`?

(4 marks)

- (b) What is the printed output of the script in Figure Q4, if the inputs are `X=10`, `Y=0`, and `Z=4`?

```
def div(A,B):
    return A//B

X=int(input('X='))
Y=int(input('Y='))
Z=int(input('Z='))
try:
    print('X div Z is ',div(X,Z))
    print('X div Y is ',div(X,Y))
    print('Z div X is ',div(Z,X))
except:
    print('Something failed!')

print('All done')
```

Figure Q4

(6 marks)

- (c) The script shown in Figure Q4 cannot distinguish and report which of the three function calls to `div` actually raise a `ZeroDivisionError` exception, if any. Describe how we can modify the script to identify and print out the related `div` function call when such an exception is raised.

(6 marks)

- (d) The Python statement

```
Myscript = open( __file__ , 'r' )
```

allows a script to open its own source code as a text file. Show how to modify the script in Figure Q4 to print out the source code of the script in the event of any `ZeroDivisionError` exception.

(9 marks)

END OF PAPER

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:

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

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