香港都會大學
HONG KONG
METROPOLITAN
UNIVERSITY

COMPS258

Computer Programming and Problem Solving

PRACTICE ASSIGNMENT

Exercises and Practices

QUESTION BOOK

The following is just to show the content that is similar to what you will see in the Take-Home Assignment. It is not relevant to this Practice Assignment.

Important Notes about the Take-Home Assignment

- This assignment is open book. Some questions are challenging, normally no perfect answer will be expected. You should consider the time is never enough. Proficiency and efficiency are part of the assessment.
- The plagiarism rules of the university are applied. The submitted work must be original, that is, your own. The students will be disqualified if plagiarised materials are submitted. For examples, students should not aid or attempt to aid another student or receive or attempt to receive aid from another person, or interact on sharing platforms, or copy a part or the whole answers from a source. Students committed plagiarism will be penalized.
- You should constantly check your email account (sxxxxxxx@hkmu.edu.hk) for announcements.
- The assignment is available for download at 09:00 on the scheduled date. The submission time is 17:00. No late submission is allowed.

Important Notes about Working on the Assignment

- You should answer in English. Show steps in your answers that involves calculation.
- Write down your assumptions to supplement the information of a question if needed.
- Over-verbose answers and answers with contradictory content will be penalised.
- You should download the **Answer Book** Word template file. Type in or copy all your answers into the designated space in the Answer Book.

Notes on Programming

- Your answers should be based on the Python programming language version 3.6 or above.
- No input validation and no parameter validation is needed unless specified otherwise. No program header is needed.
- Make assumptions if needed and write down your assumptions as comments.

Submission Method

- You should only submit the **Answer Book** in PDF format with your answers entered in the answer boxes.
    - Optionally you may submit the Answer Book in Word format as a backup.
- You are normally allowed to submit only once before the end of the prescribed deadline for submission at the OLE.

Good Luck Students!

NOTES ABOUT THIS PRACTICE ASSIGNMENT

The aim of these exercises and practices is to show you the format of the short and long questions in the take-home assignment so that you can be better prepared.

The sample questions shown here have no relation to the questions in the real assignment paper, both in content and in difficulty level. The real paper is open-book and may contain more open-ended and challenging questions which are not shown here. The open-ended questions may take you more time (of course some may think otherwise).

The most challenging open-ended problem solving questions are not included in this practice assignment.

You should answer in the Answer Book in the actual take-home assignment

Question 1 Program Comprehension and Basic Concepts

(a) Write down any two input devices.

[2]

(b) Write a complete program that reads in an integer with a suitable output message, store it in a variable, add 1 to the variable, and then print the variable out to the screen.

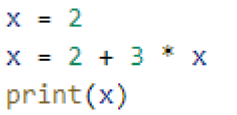
[2]

(c) According to the rules of variable naming, a name consisting of only numerals (i.e. digits), such as **`1234`**, cannot be a valid name. Explain the reason.

[2]

(d) What is the output when the following program is executed?

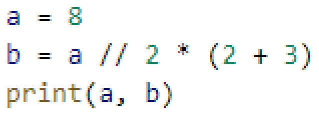
[1]



```
x = 2
x = 2 + 3 * x
print(x)
```

(e) What is the output when the following program is executed?

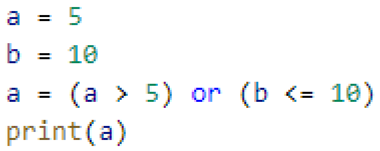
[1]



```
a = 8
b = a // 2 * (2 + 3)
print(a, b)
```

(f) What is the output when the following program is executed?

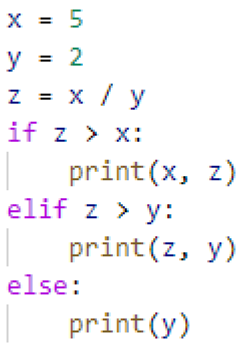
[1]



```
a = 5
b = 10
a = (a > 5) or (b <= 10)
print(a)
```

(g) What is the output when the following program is executed?

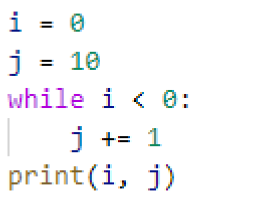
[1]



```
x = 5
y = 2
z = x / y
if z > x:
    print(x, z)
elif z > y:
    print(z, y)
else:
    print(y)
```

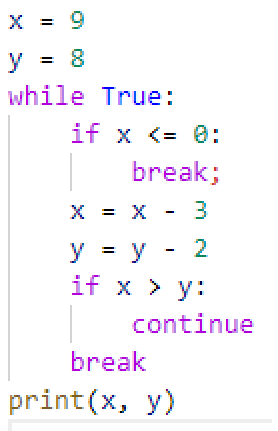
(h) What is the output when the following program is executed?

[1]



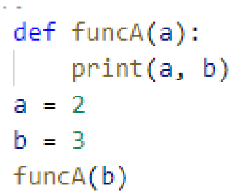
```
i = 0
j = 10
while i < 0:
    j += 1
print(i, j)
```

(i) What is the output when the following program is executed? [1]



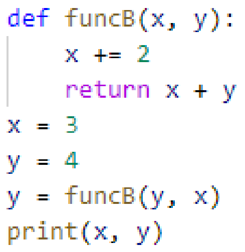
```
x = 9
y = 8
while True:
    if x <= 0:
        break;
    x = x - 3
    y = y - 2
    if x > y:
        continue
    break
print(x, y)
```

(j) What is the output when the following program is executed? [1]



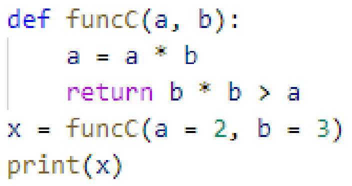
```
def funcA(a):
    print(a, b)
a = 2
b = 3
funcA(b)
```

(k) What is the output when the following program is executed? [1]



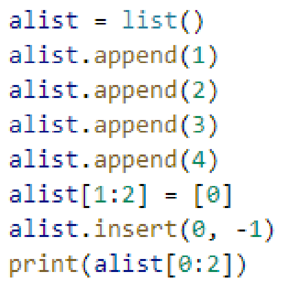
```
def funcB(x, y):
    x += 2
    return x + y
x = 3
y = 4
y = funcB(y, x)
print(x, y)
```

(l) What is the output when the following program is executed? [1]



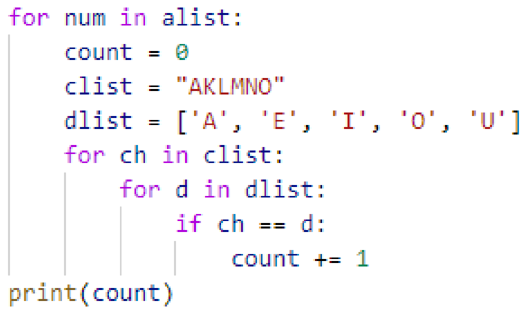
```
def funcC(a, b):
    a = a * b
    return b * b > a
x = funcC(a = 2, b = 3)
print(x)
```

(m) What is the output when the following program is executed? [1]



```
alist = list()
alist.append(1)
alist.append(2)
alist.append(3)
alist.append(4)
alist[1:2] = [0]
alist.insert(0, -1)
print(alist[0:2])
```

(n) Consider the following program.



```
for num in alist:
    count = 0
    clist = "AKLMNO"
    dlist = ['A', 'E', 'I', 'O', 'U']
    for ch in clist:
        for d in dlist:
            if ch == d:
                count += 1
print(count)
```

(i) What is the output when the following program is executed?

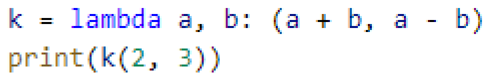
(ii) How many times will the **if** statement executed after the execution is completed?

(iii) Explain the difference between the data stored in the variables **clist** and **dlist**.

[2]

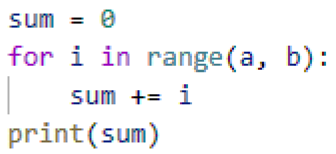
(o) What is the output when the following program is executed?

[1]



```
k = lambda a, b: (a + b, a - b)
print(k(2, 3))
```

(p) Convert the following for structure into an equivalent while structure.



```
sum = 0
for i in range(a, b):
    sum += i
print(sum)
```

[1]

Question 2. Programming Concepts and Exercises

(a) Explain the difference between integer division and regular division in Python programming with one or more examples.

[2]

(b) Write a complete program that calculates the number of free cans of a soft drink given to customers in a promotion campaign. The number of free cans is calculated from the cans of the soft drink purchased. The details are described below.

- One (1) free can is given for every 5 cans purchased.
- Two (2) extra free cans are given if purchasing 20 cans or more.

For example, if 10 cans are purchased, 2 free cans are given. If 20 cans are purchased, 6 free cans are given. The program should read in the number of cans purchased, perform the calculation, and then print the result to the screen.

[3]

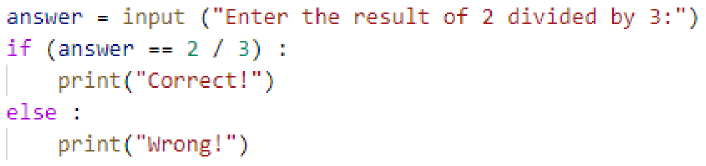
(c) Write a complete program that reads in a month as a number (e.g. 1 for January; 2 for February, etc.) and prints out the number of days in that month. Assume that the year is not a leap year (so February has 28 days). Print an error message if the input month is out of the valid range.

[2]

(d) Improve the input validation to your solution for Question 2(c) above. If the month input is invalid, in addition to printing the error message, the user is asked to re-enter.

[2]

(e) Consider executing the following program that is a simple question on arithmetic.



```
answer = input ("Enter the result of 2 divided by 3:")
if (answer == 2 / 3) :
    print("Correct!")
else :
    print("Wrong!")
```

The program prints “Wrong!” if the input is “0.6666”, “0.66666”, or even “0.666666” which are supposed to be correct. Suggest the reasons.

[2]

(f) Write a program that should read in an integer (size) and then print a square of the given size and of a pattern shown in the following examples. The examples below show two executions of the same program. The first example shows a pattern of size 5 and the second example shows one of size 8.

```
Enter size: 5
*****
*   *
*   *
*   *
*****
```

```
Enter size: 8
********
```

```
*      *
*      *
*      *
*      *
*      *
*      *
********
```

[2]

(g) An old carpenter makes furniture using British length measurements in yards, feet and inches. He would like to have a function that accepts a length measurement in meters and centimeters then converts and returns the equivalent length in yards, feet, and inches so that he can understand the requirements from the young customers using Standard International (SI) length measurements in meters and centimeters.

The function has input parameters meters that should be an integer and centimeters that should be a float. The 3 return values from the function are yards (integer) and feet (integer) and inches (float).

The conversion rate is given in the following.

- 1 meter = 100 centimeters
- 1 inch = 2.54 centimeters
- 1 foot = 12 inches
- 1 yard = 3 feet

Complete the following function.

```
def convert (meters, centimeters):
```

[2]

(h) Write a program that converts a distance from miles to kilometers. The program should perform the following tasks in the given order.

- Print a message to ask for a distance in miles.
- Perform the conversion. (Hint: Distance (in kilometer) = Distance (in miles) * 1.61)
- Print the result of the conversion to the screen.

[2]

(i) Complete the following function that returns True if all numbers in a list between the starting index (in parameter `startindex`) and the ending index (in parameter `endindex`) are all negative (below zero). It returns False if otherwise. Both indices are inclusive, that is, the numbers to check include the `startindex` and the `endindex`.

```
def isNegative(numlist, startindex = 0, endindex = None):
```

Notes

- The list is given in the parameter `numlist`
- The parameter `startindex` has a default value 0. If this parameter is not given, then the range of checking starts from the first number in the list. Assume that this parameter is valid.
- The parameter `endindex` has a default value None. If this parameter is not given or is larger than the end of the list, then the range of checking ends at the end of the list.
- The function should return `None` if `endindex` is smaller than `startindex`.

[3]

Question 3. Data Structures and Algorithms

(a) In the modular programming approach of software development, designing modules is an important task. Write down any two characteristics of a suitable module and brief explain them.

[2]

(b) Chris and Doris have each developed a sorting algorithm. Chris' sorting algorithm is implemented in the function `sortingChris`, and Doris' is implemented in the function `sortingDoris`. They are interested in comparing the performance of their sorting algorithms. Doris suggests counting the number of comparisons between data that have happened during sorting. Chris also suggested testing their algorithms with different data sizes. The following shows the evaluation results of the two algorithms.

Data sizes	***5000***	***10000***	***20000***	***40000***
`sortingChris`**: Number of comparisons**	999	3999	15999	63999
`sortingDoris`**: Number of comparisons**	2000	4000	8000	16000

[4]

(i) Explain why Chris suggested testing their algorithms with different data sizes.

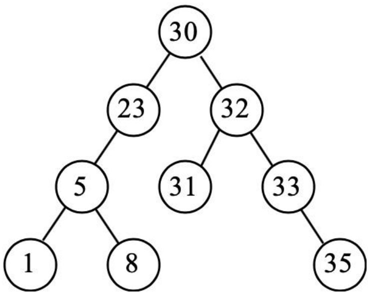
(ii) Explain why Doris suggested counting the number of comparison as a method to compare the performance of the two algorithms.

(iii) Based on the answers in the above parts, comment on the performance of the two algorithms.

(iv) Estimate the worst-case time complexity of the two sorting algorithms in Big-O notation.

(c) Consider the following binary tree. Answer the questions in the table.

[4]



(i) Is traversing a binary tree same as searching a binary tree? Explain your answer
(ii) List the nodes visited in pre-order traversal
(iii) List the nodes visited in in-order traversal
(iv) List the nodes visited in post-order traversal
(v) Write down the nodes in the right branch of 32
(vi) Write down the height of the tree
(vii) Is it a binary search tree? Justify your answer.

(d) Use binary search to search for the value of 40 in the following array of 13 integers sorted in ascending order. The array elements are numbered from 0 to 12. The middle pivot element is derived from the current number of elements divided 2, rounded down.

[3]

Show your working at every step and explain concisely what you have done. The first step is done as an example.

11	22	30	33	40	44	55	60	66	77	80	88	99

Pass 1 Begin = 0 End = 12 Mid = 6 Not Found

11	22	30	33	40	44	**55**						

(e) Assume that you have information and data about animals.

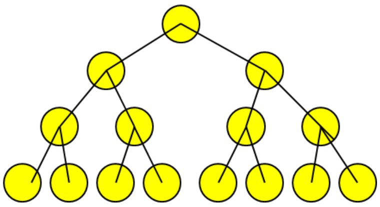
[5]

```
Cat Dog Fly Ant Pig Fish Mouse
```

(i) Build a binary search tree based on the following sequence of animal names as the keys. Begin with an empty binary tree. The names are then added to the binary search tree in the same order as the sequence.

The alphabetical order is used for comparison of names. Draw the final binary tree.

(ii) The following is known as a perfect binary tree. All leaf nodes are on the same level and all non-leaf nodes have two children. Explain why this binary search tree facilitates key searching.



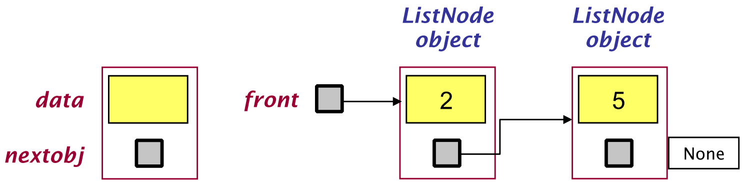
(iii) In many real situations new data are added to a binary search tree continuously. Is there a way to ensure that the built binary search tree is always a perfect binary tree? Justify your answer.

(iv) Anders asks you whether hashtable (or dictionaries) can be used to store the information about animal. Explain how it can be done by referring to the key operations of hashtables.

(v) Is using a hashtable is better than binary search tree in the aspect of performance. Comment on whether you agree and suggest how to ensure the hashtable is at the best performance.

(f) Given the following Python class definition, build a linked list as described in the figure below. The variable front is holding the address of the first node of the linked list.

```
class ListNode:
    def __init__(self, data):
        self.data = data
        self.nextobj = None
...
```



Write a program that (i) creates the above linked list based on the ListNode definition, and (ii) traverses the linked list and prints the data from front to end.

[2]

Question 4. Writing Larger Programs

(a) Complete the following function that processes a list of numbers (given in the parameter numlist). Specifically, for every odd number in the list, it is added by 1 to change it into an even number. The function should return how many odd numbers have been changed.

```
def add1Odd(numlist);
```

[2]

(b) The following table lists several ways to call the function defined in part (b) above. For each function call, write down the return value if there is no error, or write down the exception raised if there is an error.

[2]

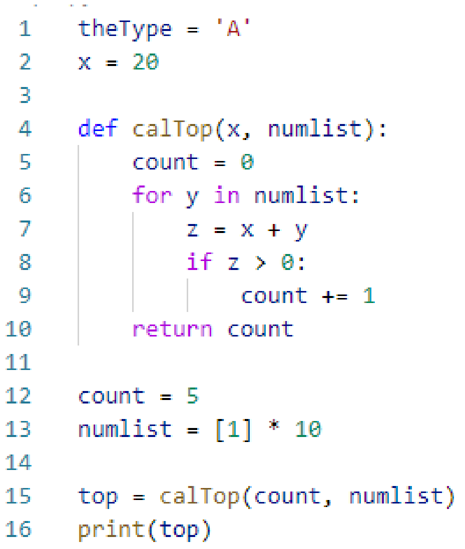
Function calls	The return value or the error occured
`add1Odd([1, 2, 3, 4])`	
`add1Odd(['A', 'B'])`	
`add1Odd(1)`	
`add1Odd()`	

(c) The following function definition is a simplified version of Question 2(i). For this question, the implementation of this function should be based on **recursion**.

```
def isNegative(numlist, startindex = 0):
```

[3]

(d) Consider the following program.



```
theType = 'A'
x = 20

def calTop(x, numlist):
    count = 0
    for y in numlist:
        z = x + y
        if z > 0:
            count += 1
    return count

count = 5
numlist = [1] * 10

top = calTop(count, numlist)
print(top)
```

[4]

(i) In the following space, list the variables that are created in the global scope (i.e. global variables) in the program.

(ii) For each of the following variables or functions, write down the line number where the variable or function is created.

Variables	Lines
`count` used on line 9	
`x` used on line 7	
`numlist` used on line 15	
`calTop` used on line 15	

(iii) Describe when a local variable is created or allocated during program execution.

(iv) Consider if the Python programming language supports only global variables. Variable declared inside and outside of function definitions are all global variables. Discuss the impact on the convenience in programming.

(e) Complete the following recursive function for evaluating the sum of the following sequence for $n > 0$

$$k = \frac{1}{1} + \frac{1}{3^2} + \frac{1}{5^2} + \frac{1}{7^2} + \ldots + \frac{1}{(2n-1)^2}$$

The number of terms n may be zero or more. The result should be 0 for $n <= 0$.

```
def evaluateSequence(n):
```

You must use recursion in your solution. Otherwise, zero mark may be awarded.

[3]

(f) Complete the following function that reduces any consecutive space sequence into a single space character in a string. The function accepts the string in the parameter named **`text`**. The function should return the filtered string.

```
def removeConSpaces(text):
```

Hint: use operator + to add two strings together.

Original String	Filtered String
`"   Programming  is fun  "`	`" Programming is fun "`

The spaces in the original string are highlighted as squares.

Original String	Filtered String
`"□□□Programming□□is□fun□□"`	`"□Programming□is□fun□"`

[3]

(g) The slicing operation can be applied to lists, strings, and tuples. Add remarks to explain the examples of slicing in the table, and then explain how slicing works. The variable str is a string.

```
str = 'Programming'
```

[3]

Slicing examples	Remarks
`str[1:4]`	
`str[:-2]`	
`str[1::2]`	
`str[::-1]`	

Question 5. Advanced Programming and Problem Solving

(a) Write a Python program that can report statistical summary about employment of population in different districts and areas in Hong Kong. The program should read from a file named district.txt, which contains the following information.

- Districts and areas
- Number of employees
- Number of employers
- Number of students
- Number of retired
- Number of other people

The first few lines of the file district.txt are extracted and shown below. The first line contains header information (non-data). Each line to follow contains one district given its name (at most 63 characters), and then followed by the five raw statistics as integers. The fields are separated by a comma.

```
District,Employees,Employers,Students,Retired,Others
Central & Western - Chung Wan,7576,1042,2156,1875,3154
Central & Western - Mid Levels East,9221,1361,3037,2480,2926
Central & Western - Castle Road,9336,1460,3033,2091,2633
Central & Western - Peak,6505,766,1606,2071,2725
Wan Chai - Hennessy,6505,766,1606,2071,2725
Central & Western - Oi Kwan,6232,737,1919,2696,2651
Central & Western - Canal Road,7607,982,2021,2913,3083
...
```

The name of a district is made up by two parts separated by a hyphen ('-'), which are the large district and a small district. For example, the district name "Wan Chai - Oi Kwan" means that the large district is "Wan Chai" and the small district is "Oi Kwan".

Complete a program to process the file and print the following statistical summary to the screen (in an appropriate format):

- The total number of districts in the raw data set.
- The total number of employers in all districts.
- The total number of retired people in the "Wan Chai" large district.
- The name of the district with the highest number of students.

[6]

(b) Betsy is reading the implementation of the Stack ADT in Unit 7. This implementation is based on the Python list class. The following shows the implementation.

[6]

```
#File Name: pyListStack.py
class Stack:
    def __init__(self):
        self.items = []

    def isEmpty(self):
        return self.items == []

    def push(self, item):
        self.items.append(item)

    def pop(self):
        assert not self.isEmpty(), "Cannot pop from an empty stack"
        return self.items.pop()

    def peek(self):
        assert not self.isEmpty(), "Cannot peek at an empty stack"
        return self.items[-1]

    def size(self):
        return len(self.items)
```

(i) Test the implementation by writing code to create a Stack. Then write a loop to push 10 random integers (in the range 0 to 9 inclusive) to the Stack. After that, pop the stack 10 times to get back the integers and print them out.

(ii) Explain the relation between the order of data pushed to the stack and that pop'ed from the stack.

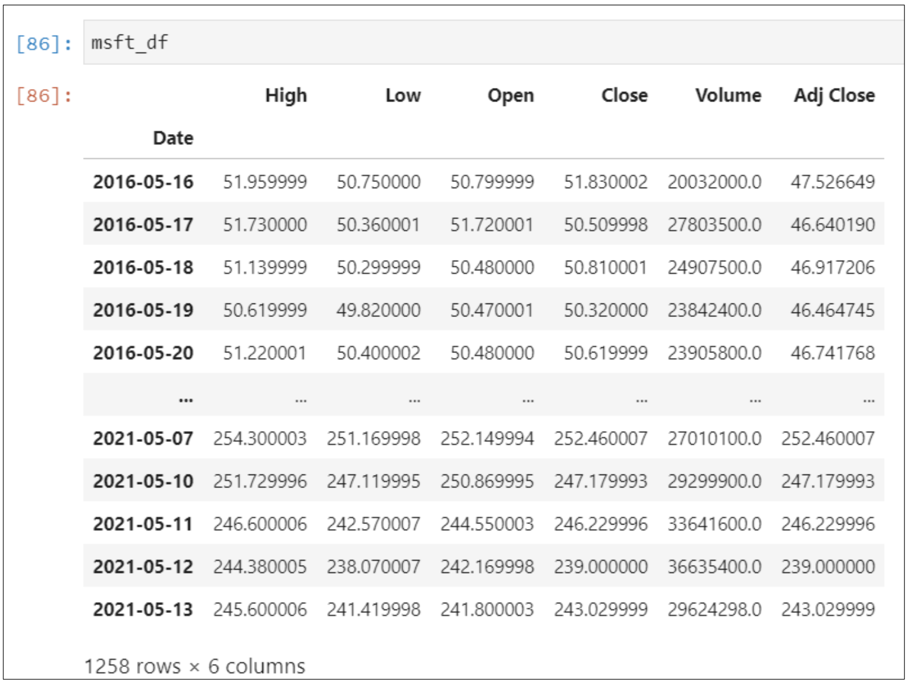
(iii) Explain the meaning of `assert` in the implementation above.

(iv) Complete the following function for the Stack class that swaps the top two data items in the stack. The function should do nothing if there are fewer than 2 data items.

(v) Write down the worst-case time complexity for your implementation of the function **`swap`**. Justify your answer.

(c) This question is about data analysis with Python, in particular, the pandas module and the dataframe.

Consider that you are given a partially completed Python program (in a Jupyter Notebook) that has loaded data into a dataframe in the variable msft_df. The dataframe contains 5 years of stock information for Microsoft Corporation (stock code: 'MSFT').



```
[86]: msft_df

[86]:
                  High         Low        Open       Close      Volume   Adj Close
       Date
 2016-05-16    51.959999   50.750000   50.799999   51.830002  20032000.0   47.526649
 2016-05-17    51.730000   50.360001   51.720001   50.509998  27803500.0   46.640190
 2016-05-18    51.139999   50.299999   50.480000   50.810001  24907500.0   46.917206
 2016-05-19    50.619999   49.820000   50.470001   50.320000  23842400.0   46.464745
 2016-05-20    51.220001   50.400002   50.480000   50.619999  23905800.0   46.741768
        ...          ...         ...         ...         ...         ...         ...
 2021-05-07   254.300003  251.169998  252.149994  252.460007  27010100.0  252.460007
 2021-05-10   251.729996  247.119995  250.869995  247.179993  29299900.0  247.179993
 2021-05-11   246.600006  242.570007  244.550003  246.229996  33641600.0  246.229996
 2021-05-12   244.380005  238.070007  242.169998  239.000000  36635400.0  239.000000
 2021-05-13   245.600006  241.419998  241.800003  243.029999  29624298.0  243.029999

1258 rows × 6 columns
```

[4]

(i) Write code to get the closing price on 2018-10-11 (that is, 11-Oct-2018).

(ii) Write code to get the 5 highest closing prices

(iii) Write code to get all the rows where the closing price is 240 dollars or higher.

(iv) Fill in the correct arguments or parameters needed for pyplot.plot() function call to produce a line plot of the closing prices

```
import matplotlib.pyplot as plt

plt.plot(                    )

plt.show()
```

(d) Chris is a book lover and he has lots of books in a mini-library at home. His books are popular among his friends that they come and borrow them. The problem is that sometimes they may forget to return the books. As a COMPS258 student, you decide to help Chris write a program that manages his books.

The book management program should allow the following features or operations:

- Add a new book record.
- Delete a book record.
- Note that a book is borrowed.
- Note that a book is returned.
- List all books that are borrowed and not returned.
- List all the books that are borrowed by a particular friend.

[4]

(i) You decide to design an ADT for a book and an ADT for the library. Write down, in the space below, the operations or functions for the two ADT to support the above features.

- Book ADT
- Library ADT

(ii) State the best data structure for implementing the Library ADT for the program, and justify your choice.

(iii) Explain the advatnage of modelling the library and the books with an ADT, rather than simply functions.

[END OF PRACTICE ASSIGNMENT]