ANGLO-CHINESE JUNIOR COLLEGE
JC1 PROMOTIONAL EXAMINATION

Higher 2

COMPUTING 9569/02

Paper 2 (Lab-based) **25 August 2022**

1 hour 30 minutes

Additional Materials: Electronic version of `text1.txt` data file
Electronic version of `text2.txt` data file
Insert Quick Reference Guide

READ THESE INSTRUCTIONS FIRST

Answer **all** questions.

All tasks must be done in the computer laboratory. You are not allowed to bring in or take out any pieces of work or materials on paper or electronic media or in any other form.

Approved calculators are allowed.

Save each task as it is completed.

The use of built-in functions, where appropriate, is allowed for this paper unless stated otherwise.

Note that up to **3** marks out of 50 will be awarded for the use of common coding standards for programming style.

The number of marks is given in brackets [ ] at the end of each question or part question.
The total number of marks for this paper is 50.

Instruction to candidates:

Your program code and output for each of Task 1 and 2 should be downloaded in a single .ipynb file. For example, your program code and output for Task 1 should be downloaded as
`TASK1_<your name>_<index number>.ipynb`

1 For this question, you are not allowed to use the `int`, `bin`, `oct` or `hex` functions.

Task 1.1

An integer is represented in binary notation by a string consisting only of `1`s and `0`s. Such a string is called a binary string.

Write program code that converts a binary string into the integer it represents. [3]

Task 1.2

ASCII can be used to encode a string written with the English alphabet. Each character is encoded to an integer from 0 to 127 inclusive. These are converted to binary digits (bits), so that each character corresponds to eight bits (one byte). The bytes are concatenated and stored as a sequence of `1`s and `0`s.

Write program code to convert a sequence of `1`s and `0`s (given as a string) into English text. [4]

Test your code by reading the file `text1.txt`, converting it to text, and writing the text into a new file, `answer.txt`. [5]

Task 1.3

To handle languages which have more characters, or multiple languages are the same time, the ASCII encoding system was extended into Unicode. In Unicode, each character is still encoded to an integer. The number of bytes used to encode the character depends on the length of the integer.

If the integer is between 0 and 127, at most seven bits are required, so zeroes are added in front to make it a complete byte.

If at most eleven bits are required, zeroes are added in front so that there are exactly eleven bits. These eleven bits are embedded into two bytes in the following way:

```
110XXXXX 10XXXXXX
```

where the `X`s represent the eleven bits of the character in order.

Similarly, if at most sixteen bits are required, the integer is encoded across three bytes in the following way:

```
1110xxxx 10xxxxxx 10xxxxxx
```

Write program code to convert a sequence of `1`s and `0`s (given as a string) into text. [9]

Test your code by reading the file `text2.txt`, converting it to text, and giving the result as output in Jupyter Notebook. [2]

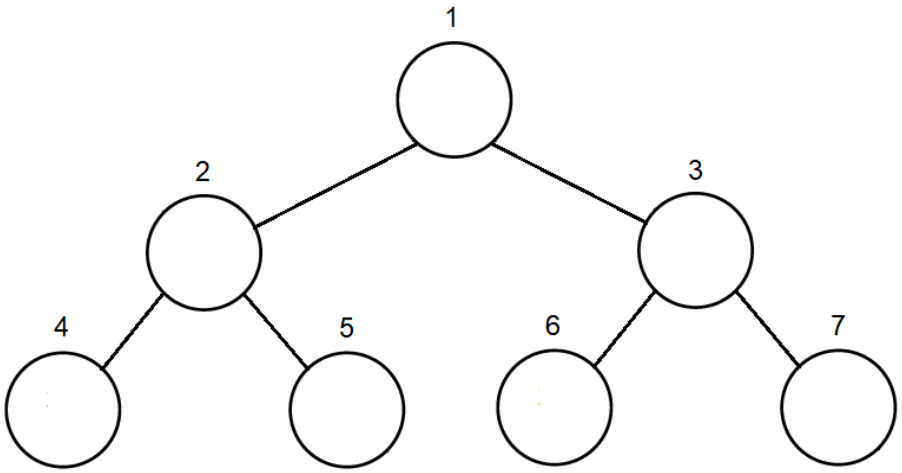
Download your program code and output for Task 1 as
`TASK1_<your name>_<index number>.ipynb`

2 An array is a simple data structure that can be used to represent a binary search tree.

Below is an example of an array of length 7. Each cell corresponds to a node in a binary search tree where data can be stored. Node 1 is always the root node.

Node	1	2	3	4	5	6	7

The array represents the following binary search tree with the nodes labelled accordingly. Study the pattern of the nodes carefully.



For this purpose, the superclass `Array` is used to implement the subclass `BST`, which stands for binary search tree.

Assume that:
- the length of the defined array is always enough to accommodate the nodes to be inserted, i.e. **much** larger than the number of the nodes to be inserted into the binary search tree;
- each node in the binary search tree contains either zero or a positive integer only.

Task 2.1

The table below shows the properties and methods of the superclass `Array`.

Array
- array: LIST - length: INTEGER
+ Constructor(INTEGER) + size(): INTEGER + get(INTEGER): DATA + set(DATA, INTEGER)

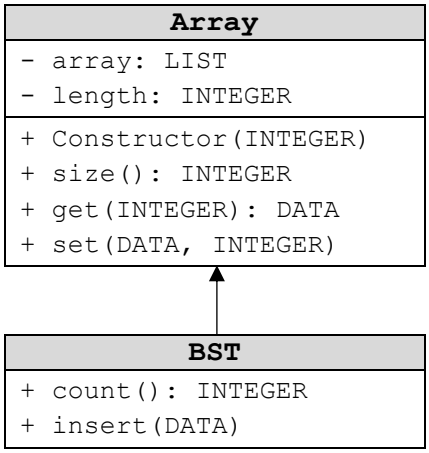
- `size()` returns the length of the array.
- `get(INTEGER)` takes in a value and returns the `DATA` stored in the node numbered `INTEGER`.
- `set(DATA, INTEGER)` stores `DATA` into the node numbered `INTEGER`.

Write program code for the superclass `Array` that allows the user to create an array of a defined length. Initialise the array with appropriate values.

Assume that the input for all the methods, if any, is always valid. [5]

Task 2.2

The diagram below show the superclass `Array`, as well as the subclass `BST` with two methods to be defined.



- `count()` returns the number of filled nodes available in the binary search tree.
- `insert(DATA)` stores `DATA` in the appropriate node in the binary search tree.

Using appropriate inheritance, write program code for the subclass `BST`.

Assume that the input for all the methods, if any, is always valid. [7]

Task 2.3

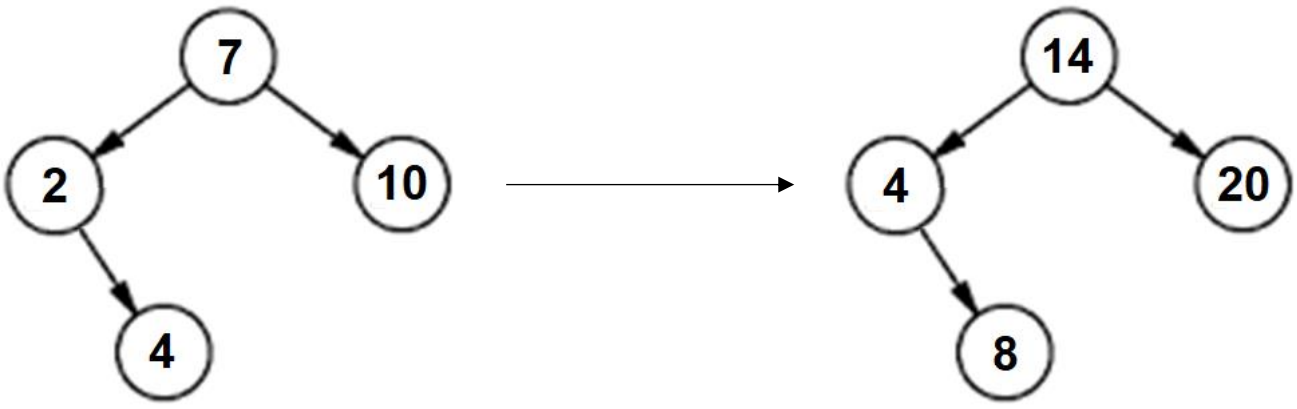
With an appropriate traversal mode, a copy of a binary search tree can be created.

Modify your `BST` class in **Task 2.2** to include the appropriate traversal mode that returns the sequence of data in an appropriate format. [6]

Hence, write a function `double_copy(tree)` that:

- takes a `BST` object `tree`;
- returns a new `BST` object that is the exact copy of `tree`, but with double the value in each node. [3]

For example,



`tree` `double_copy(tree)`

Test your function by:

- creating an empty binary search tree of length 100;
- inserting 6 appropriate nodes into the binary search tree such that:
  - a boundary test case is covered,
  - the binary search tree has a height of 4;
- creating a modified copy of the binary search tree as described above. [3]

Download your program code and output for Task 2 as
`TASK2_<your name>_<index number>.ipynb`