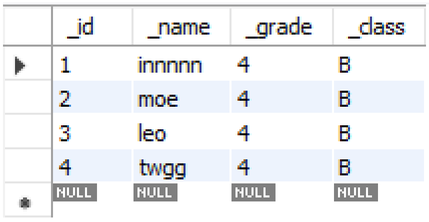
LAB 1120 AOOP

1. Practice of Database (MySQL Workbench)

According to the lecture, you should complete the following requirements:

1) Create a new schema.
2) Create a new table with four columns, id, name, grade, class. The private key is id.
   a) id: int
   b) name: char(20)
   c) grade: int
   d) class: char(1)
3) Add two data manually into schema:
   a) ”1”, “innnnn”, “4”, “B”
   b) “2”, “moe”, “4”, “B”
4) Add another two data from “csv”.
5) You have to use SQL command to complete this problem except for loading data from “csv”.

Sample Output

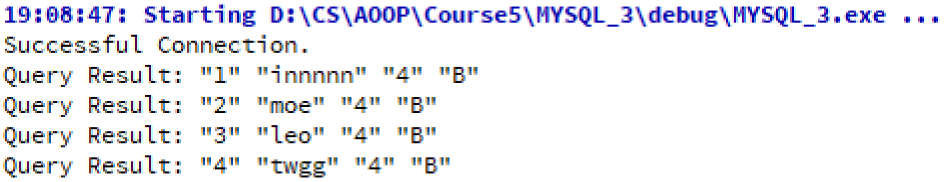


	_id	_name	_grade	_class
▶	1	innnnn	4	B
	2	moe	4	B
	3	leo	4	B
	4	twgg	4	B
✱	NULL	NULL	NULL	NULL

2. Practice of Database (Connection with Qt)

According to the lecture, you should write a Qt widgets application to get the data from database which you create in Problem 1.

Sample Output

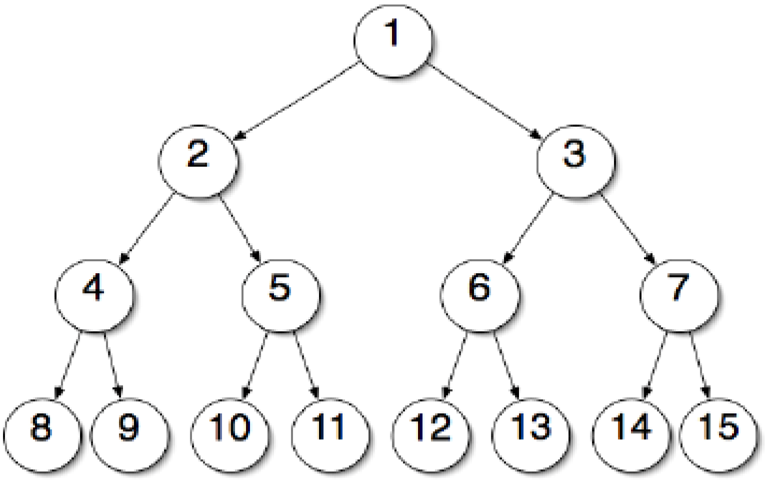


```
19:08:47: Starting D:\CS\AOOP\Course5\MYSQL_3\debug\MYSQL_3.exe ...
Successful Connection.
Query Result: "1" "innnnn" "4" "B"
Query Result: "2" "moe" "4" "B"
Query Result: "3" "leo" "4" "B"
Query Result: "4" "twgg" "4" "B"
```

3. Largest Root (MyTree)

According to the diagram below, find the largest value among these subtrees. (圖中數字代表節點的編號)

Definition of subtree: find an arbitrary node, this node can construct a tree with all of its child nodes.



Each node contains two values, one is its own node value, and another is the value of the subtree with this node being the root.

The value of each subtree is calculated by adding up the node value of its root and **half of the sum** of the values of its two child subtrees.

You need to find out which subtree has the maximum value and what its value is.

Sample input

10 20 30 40 50 60 70 80 90 100 110 120 130 140 150

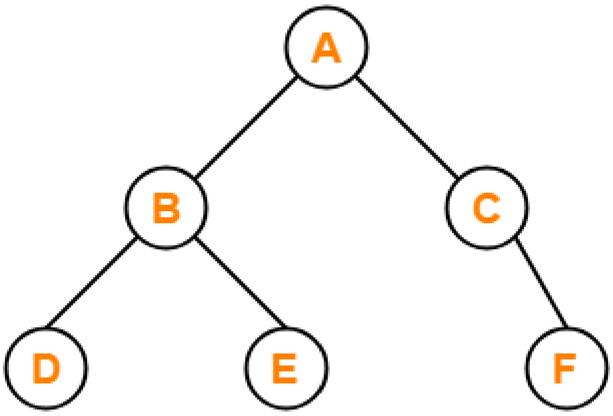
//15 values will be inputted, and the index of each input value correspond to the number of nodes, and these values aren't necessary to be positive

Sample output:

3:230.000

// display output value as a normal number (in comparison with scientific notation), and show three digits after decimal point

4. Find Structures of Trees (MyTree)



There are three ways to describe trees:

1) Pre-order : visit root -> go left -> go right (ABDECF)
2) In-order: go left -> visit root -> go right (DBEACF)
3) Post-order: go left -> go right -> visit root (DEBFCA)

Your job is to find the tree's structure according to the description of it.

Sample input (characters of inputs will be a-z or A-Z)

ABDECF DBEACF

// input are Pre-order description first, then In-order description

Sample output

DEBFCA

// output the Post-order description of the tree that construct from input.

Hint:

You can use the structure below in your "MyTree" class

```
struct node {
  struct node *left;  //left child node
  struct node *right; //right child node
  string keyStr;
  int keyInt;
}
```