**INSTRUCTIONS - PRACTICAL EXAM – CSD201**

Read the instructions below carefully before start coding.

**======================================================================**

**Students are ONLY allowed to use:**

* Software tools must be used: **NetBeans IDE 8.x** and **Java JDK 1.8**.
* His / her own study materials like presentation slides, notes, sample codes, program examples, electronic books stored on his / her computer only.
* For distance learning: Google Meet, Hangout (for Exam Monitoring Purpose).

**Instructions**

* Step 1: run “Clean and Build Project” (Shift+F11) to create **dist** folder and **.jar** file.
* Step 2: Prepare to submit answer:
  + For each question (e.g., question **1**), please create two sub-folders: **run** and **src**.
  + Copy \*.**jar** file into **run** folder
  + Compress source code into **.zip**, then copy .**zip** file into **src** folder.
* Step 2: Submit solution for each question:
  + Choose question number (e.g., **1**) in PEA software, and then attach corresponding solution folder (e.g., **1**). Click Submit button to finish submitting this question.
  + 

**Notes**

* **Do not use accented Vietnamese** when writing comments in programs.
* Solutions will be marked by Automated Marking Software.
* **If at least one of the above requirements is not followed, the exam will get ZERO.**

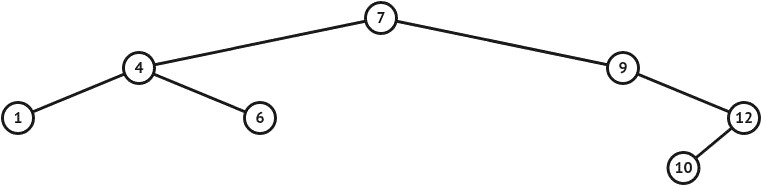
**======================================================================**

**Question 1: (3 marks)**

Write a program to build a binary search tree by inserting N (1 ≤ N ≤ 100) integer values into the BST one by one. Note that the nodes have no duplicate values.

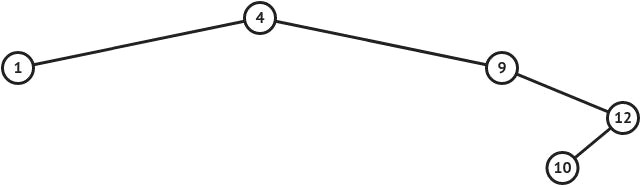
Your task is delete M (1 ≤ M < N) nodes of the BST and display the pre-order and post-order traversing of the BST.

*For example,*

* Create a BST(1) tree by successively adding to the tree N integer values as follows: 7, 9, 4, 1, 12, 6, 10.
* The pre-order traversing of the BST(1) tree is: 7, 4, 1, 6, 9, 12, 10.
* The post-order traversing of the BST(1) tree is: 1, 6, 4, 10, 12, 9, 7.

*Figure 2. The BST(1) that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one*

After deleted two nodes 6 and 7 of the BST(1):

* The pre-order traversing of the BST(2) tree is: 4, 1, 9, 12, 10.
* The post-order traversing of the BST(2) tree is: 1, 10, 12, 9, 4.

*Figure 3. After deleted two nodes 6 and 7 of the BST(1), we get the BST(2)*

**The input**: are stored in the ***ex02\_input.txt*** text file:

* The first line contains a positive integer N (1 ≤ N ≤ 100) which is the number of integer values to insert into the BST.
* The second line containing N integers that will be inserted into the BST one by one, each number separated by at least one space.
* The third line contains a positive integer M (1 ≤ M < N) which is the number of values to be deleted.
* The fourth line containing ***M*** integers that will be deleted one by one, each number separated by at least one space.

**The output:** the results need to be saved to the ***ex02\_output.txt*** text file:

* The first line contains the sequence of numbers representing the pre-order traversing of the BST(2).
* The second line contains the sequence of numbers representing the post-order traversing of the BST(2).

|  |  |
| --- | --- |
| *Sample Input 1* | *Sample Output 1* |
| 7  7 9 4 1 12 6 10  2  6 7 | 4,1,9,12,10  1,10,12,9,4 |

|  |  |
| --- | --- |
| *Sample Input 2* | *Sample Output 2* |
| 9  10 7 6 2 18 13 2 6 4  4  10 7 6 13 | 4,2,18  2,18,4 |

**Question 2: (4 marks)**

Write a program to build an **undirected** graph by giving adjacency matrix.

Your task is to find degree of each vertex in the given graph.

*For example,*

* Giving the adjacency matrix that used to represent undirected graph as follow:

|  |  |
| --- | --- |
| 5  0 1 1 1 0  1 0 1 0 0  1 1 0 0 0  1 0 0 0 1  0 0 0 1 0 | *Figure 5. The undirected graph that created by giving adjacency matrix*  0(3),1(2),2(2),3(2),4(1) |

**The input**: are stored in the ***degree\_input.txt*** text file:

* The first line contains a positive integer N (1 ≤ N ≤ 100) which is the number of vertex of undirected graph.
* The next ***N*** line, each line containing ***N*** integers that represent the adjacency matrix.

**The output:** the results need to be saved to the ***degree\_output.txt*** text file:

* The first line contains the number of the cut edge.
* The second line contains the list of the cut edge, each number separated by one comma.

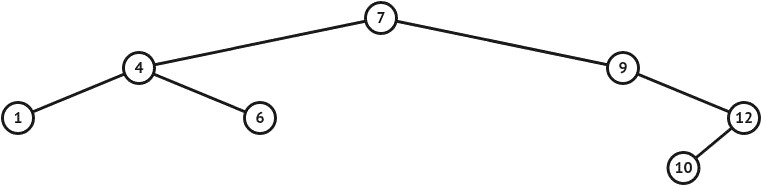
|  |  |
| --- | --- |
| *Sample Input* | *Sample Output* |
| 5  0 1 1 1 0  1 0 1 0 0  1 1 0 0 0  1 0 0 0 1  0 0 0 1 0 | 0(3),1(2),2(2),3(2),4(1) |
| 7  0 1 0 0 0 0 0  1 0 1 0 0 0 0  0 1 0 1 1 0 0  0 0 1 0 0 0 0  0 0 1 0 0 1 0  0 0 0 0 1 0 1  0 0 0 0 0 1 0 | 0(1),1(2),2(3),3(1),4(2),5(2),6(1) |

**Question 3: (3 marks)**

Write a program to build a binary search tree by inserting N (1 ≤ N ≤ 100) integer values into the BST one by one. Note that the nodes have no duplicate values.

Your task is to display the DFS and BFS traversing of the BST.

*For example,*

* Create a BST tree by successively adding to the tree N integer values as follows: 7, 9, 4, 1, 12, 6, 10.
* The DFS traversing of the BST tree is: 7,4,1,6,9,12,10
* The BFS traversing of the BST tree is: 7,4,9,1,6,12,10

*Figure 1. The BST that created by inserting 7, 9, 4, 1, 12, 6, 10 one by one*

**The input**: are stored in the ***ex08\_input.txt*** text file:

* The first line contains a positive integer N (1 ≤ N ≤ 100) which is the number of integer values to insert into the BST.
* The second line containing N integers that will be inserted into the BST one by one, each number separated by at least one space.

**The output:** the results need to be saved to the ***ex08\_output.txt*** text file:

* The first line contains the sequence of numbers representing the DFS traversing of the BST.
* The second line contains the sequence of numbers representing the BFS traversing of the BST.

|  |  |
| --- | --- |
| *Sample Input 1* | *Sample Output 1* |
| 7  7 9 4 1 12 6 10 | 7,4,1,6,9,12,10  7,4,9,1,6,12,10 |

|  |  |
| --- | --- |
| *Sample Input 2* | *Sample Output 2* |
| 9  10 7 6 2 18 13 2 6 4 | 10,7,6,6,2,2,4,18,13  10,7,18,6,6,13,2,2,4 |