

Dharmsinh Desai University, Nadiad
Faculty of Technology
Department of Computer Engineering
B. Tech – CE, Semester: III
Subject: Data Structure and Algorithm

General Instructions:

- First things first, we take plagiarism very seriously for this course. So please do not copy other's work. If anyone is found guilty of plagiarism, very strict action would be taken. As you have to submit your work online, its easy to detect plagiarism on soft copy. So be aware!
- Unindented code really looks ugly and is hard to follow program logic. So please make sure that your program is properly indented before submission.
- Names of the variables should be meaningful and should reflect its usage. Do not use variable names like a, b, c.
- Divide your logic into multiple small functions whenever appropriate, do not write entire logic in single function.
- Assume all the inputs to your program will always be within given range.
- Test your program with different test cases (different inputs) and try to make it robust before submission.
- Submit solution for problem statements on moodle before deadline for respective lab work. No excuse would be entertained after deadline.
- Name of the program file should be RollNoOnlyThreeDigits_ProgramNoTwoDigits.c. E.g 001_02.c – it means that this is the solution from roll no. CE001 for second program. If you do not follow this naming convention, your submission will not be considered.
- **DO NOT** use statements like `printf("Enter number: ")` or `printf("Output is: ")`. Output file should be clean as explained in problem statement, without any clutter.
- Ask questions, whenever in doubt. If its related to programming, do some work to resolve it by yourself first, if you can not resolve it then ask for help.

Lab 8: Binary Tree Traversal and Binary Search

Mandatory:

1. Write a program to create binary tree and perform pre-order, in-order and post-order traversal.

Input format:

- First line would contain number **n**, which represents number of nodes in the tree (1 to n).
- Next **n** lines would contain information regarding children of each node. Each line would have three numbers (separated by space), first number in each of these lines would represent node number, second number would represent its left child and third number would represent its right child.
- Second line in the input would always be representing root node and its children.
- If -1 appears as value of left child then it means that there is no left child. Same applies to right child as well.

Output format:

- Output should be consisting of three lines
- First line will contain **n** numbers (separated by space), representing pre-order traversal of the input binary tree.
- Second line will contain **n** numbers (separated by space), representing in-order traversal of the input binary tree.
- Third line will contain **n** numbers (separated by space), representing post-order traversal of the input binary tree.

NOTE: Please note that each edge in the tree is directed. You are free to use any binary tree representation of this tree in memory. You can also use either recursive or iterative algorithm to solve this problem.

Range:

$1 \leq n \leq 10000$

Example 1:

Input:

```
11
1 2 7
2 3 6
3 4 5
6 -1 -1
4 -1 -1
5 -1 -1
7 8 9
8 -1 -1
9 10 11
10 -1 -1
11 -1 -1
```

Output:

```
1 2 3 4 5 6 7 8 9 10 11
4 3 5 2 6 1 8 7 10 9 11
4 5 3 6 2 8 10 11 9 7 1
```

2. Write a program to perform binary search on given list of ordered numbers.

Input format:

- First line would contain number **n**, representing numbers in the input list.
- Second line would contain **n** numbers (separated by space), these numbers would be present in ascending order.
- Third line would contain number **m**
- Fourth line would contain **m** numbers (separated by space), these are the numbers which have to be searched from line 2 using binary search.

Output format:

- Output would be of one line having **m** binary digits (separated by space).
- Binary digit 0 means corresponding number in forth line in the input is not present in the second line in the input.
- Binary digit 1 means corresponding number in forth line in the input is present in the second line in the input.

NOTE: You are free to use any binary tree representation of this tree in memory.

Range:

1 <= **n, m** <= 10000

1 <= **number in the input list** <= 1000000

1 <= **number to be searched** <= 1000000

Example 1:

Input:

15

3 5 7 12 19 21 100 10001 30231 45656 65464 75464 546454 754645 999999

7

4 10001 6 7 13 65789 999999

Output:

0 1 0 1 0 0 1

Practice Questions:

3. Generate post-order traversal of BST from given pre-order traversal.

Input format:

- First line would contain number **n**, which represents number of nodes in the BST.
- Second line contains **n** numbers (separated by space), representing pre-order traversal of BST.

Output format:

- Output should be one line containing **n** numbers (separated by space), representing post-order traversal of BST.

NOTE: You are free to use any binary tree representation of this tree in memory.

Range:

1 <= **n** <= 10000

1 <= **each number in the BST** <= 1000000

Example 1:

Input:

9

5001 3337 2211 5 4200 11111 7777 8888 22222

Output:

5 2211 4200 3337 8888 7777 22222 11111 5001