

5Instructions:

1. Please write the program in C language.
2. Use the program name as "*A⟨assignmentNo⟩_⟨rollNo⟩.c*". For example A5_15XXXXXX.c for the program of assignment-5 submitted by 15XXXXXX. If a group is submitting the assignment, then the program name should be "*A⟨assignmentNo⟩_⟨rollNo1⟩_⟨rollNo2⟩.c*".
3. Upload the latest program in your google drive and share it to "**assignments.ds@gmail.com**". The procedure is as follows.
 - Right click the uploaded file and click "share".
 - Select "Advanced" button at the bottom right corner.
 - Give "assignments.ds@gmail.com" as "view access".
 - Untick the Checkbox for "Notify People" in order to avoid email notification.
 - Click "Ok" and then "Done".
4. The code can be deleted and submitted again with the same name, any time before 11:59 pm of the last date of submission. The latest file will be used for evaluation. Any file submitted after the deadline will not be evaluated.
5. As the file is only shared as view access to the evaluator, students are not supposed to delete the file in their own drive after the deadline, which may cause deletion of the shared file in evaluator's drive. Hence keep the files in the drive unchanged till the end of the semester. This is required to maintain the ownership of the file.
6. An Input test case file is also attached to the assignment notification email. This input can be given to the program using the following syntax.
./a.out <⟨inputFileName⟩.txt
The code snippet for reading this input is given below.
7. Please refrain from copying the code.
8. Last date of submission: **13th November 2016 11:59 pm.**

Code Snippet to read the input test case

```
#include<stdio.h>

void main(){
    int n,i;
    char c[20][100];
    scanf("%d\n",&n);

    printf("\nThe number of expressions , n = %d",n);

    for (i=0;i<n;i++){
        gets(c[i]);
        printf("\nc[%d]=%s",i,c[i]);
    }
```

```
}  
}
```

Question

Assuming a binary tree exists with each node containing a **distinct letter**(alphabet), the input file contains such tree instances with the **pre-order** traversal of the tree followed by the **in-order** traversal of the tree in the next line. Thus the first line of the input file contains the number of such instances and the second line onwards the respective pre-order and in-order traversal of the instances are listed.

Write a C program to output the **post-order** traversal of the tree instances.

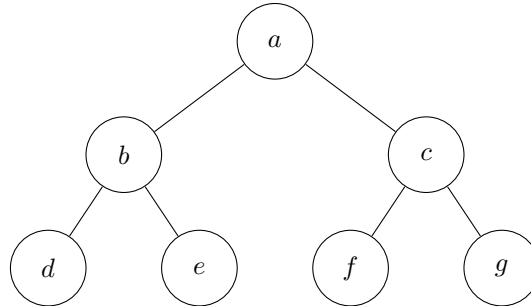


Figure 1: Assumed binary tree for the first instance in the sample input

Sample Input

```
2  
abdecfg  
dbeafcg  
kmopnqr  
ompkqnr
```

Output

```
debfzca  
opmqnrk
```