

2. An MNC finds *employees of the month* based on the reward points collected by the employees in that month. The company uses a tree data structure to store the employee ID,  $E\_id$  and reward points,  $r$  of each employee. Under each employee, at most two teams of employees are present. An employee is chosen as an *employee of the month*, if his reward points are greater than the sum of the reward points of the employees in the teams under him. Note that employees without any employee under them cannot be chosen as employee of the month. Given the employee ID,  $E\_id$  and the reward points,  $r$  of the employees, write a program to find the *employees of the month*. Your program should include the functions given below.

- `read_and_store()`: Read the employee ID and reward points of the employees from the given parenthesis notation of the binary tree, and create the tree. [2 Marks]
- `print_employees(T)`: Print the employee IDs of all the employees stored in the binary tree  $T$ , in its in-order traversal. [1 Mark]
- `find_employees(T)`: Given the tree  $T$  storing the employee details, find and print the employee IDs of all the employees selected as the *employee of the month*. **This function should run in  $O(n)$  time complexity.** [3 Marks]

### Input Format

- The input contains the Parenthesis Representation of the binary tree with employee details (a string  $E\_id$  of maximum length 100 and an integer  $r \in [0, 10^5]$ ) which is recursively defined in the following manner.
  - The string `()` represents an empty tree.
  - The string `(  $E\_id$   $r$  teamL teamR )` represents a tree whose root node is an employee with Employee ID  $E\_id$  and reward points  $r$ , teamL is the left subtree of the root node in Parenthesis Representation and teamR is the right subtree of the root node in Parenthesis Representation.

### Output Format

- The first line of the output contains the employee IDs of all the employees stored in the binary tree  $T$  in its in-order traversal, separated by a space.
  - The second line of the output contains the employee IDs of the *employees of the month* in the order in which they appear in the post-order traversal of  $T$ , separated by a space.
-

### Sample Input and Output

#### Input 1

```
( E101 10 ( E102 35 ( E103 15 ( E104 8 ( ) ( ) ) ( E105 6 ( ) ( ) )  
) ( E106 5 ( ) ( ) ) ) ( E107 13 ( E108 1 ( ) ( ) ) ( E109 7 ( ) ( )  
) ) )
```

#### Output

```
E104 E103 E105 E102 E106 E101 E108 E107 E109  
E103 E102 E107
```

#### Input 2

```
( E201 100 ( E120 27 ( E133 14 ( E114 8 ( ) ( ) ) ( E145 5 ( ) ( ) )  
) ( ) ) ( E147 13 ( E178 6 ( ) ( ) ) ( E129 15 ( ) ( E110 3 ( E151 8  
( ) ( ) ) ( ) ) ) ) )
```

#### Output

```
E114 E133 E145 E120 E201 E178 E147 E129 E151 E110  
E133 E129 E201
```