|  |  |  |  |
| --- | --- | --- | --- |
|  | Name | LeftPointer | RightPointer |
| 1 | Dodi | 5 | 2 |
| 2 | Farai | 3 | 4 |
| 3 | Elli |  |  |
| 4 | George | 6 |  |
| 5 | Ben | 7 |  |
| 6 | Celine |  |  |
| 7 | Ali |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |
| FreePointer |  | RootPointer | 1 |

Ali

Ben

Celine

Elli

George

Farai

Dodi

4

01 void TraverseTree(int Root){  
02 if Tree[Root].LeftPointer.Name!=0  
03 {  
04 TraverseTree(Tree[LeftPointer])  
05 }  
06 OUTPUT Tree[Root].Name  
07 if Tree[Root].RightPointer.Name != 0  
08 {  
09 TraverseTree(Tree[RightPointer])  
10 }  
11 }

Объясните, что подразумевается под рекурсивной функцией. Укажите номер строки из кода выше, которая показывает, что метод TraverseTree является рекурсивным.

Рекурсивная функция - функция, вызывающая саму себя

Номер строки 4 [2]

(iii) Запишите вызов, необходимый для вывода всех имен, сохраненных в дереве.

TraverseTree(1) [1]