Đã bắt đầu vào lúc	Chủ nhật, 29 Tháng mười 2023, 10:25 AM
Tình trạng	Đã hoàn thành
Hoàn thành vào lúc	Chủ nhật, 29 Tháng mười 2023, 2:52 PM
Thời gian thực hiện	4 giờ 27 phút
Điểm	2,00/2,00
Điểm	**10,00** của 10,00 (**100**%)

Câu hỏi **1**

Chính xác

Điểm 1,00 của 1,00

Class `ETNode` is used to store a node in expression tree, described on the following:

```
class ETNode {
public:
    char val;
    ETNode *left;
    ETNode *right;
    ETNode() {
        this->left = this->right = NULL;
    };
    ETNode(char val) {
        this->val = val;
        this->left = this->right = nullptr;
    }
    ETNode(char val, ETNode* left, ETNode* right) {
        this->val = val;
        this->left = left;
        this->right = right;
    }
};
```

Where `val` is the value of node (either an operand (`'0'-'9'`) or an operator (`+,-,*`)), `left` and `right` are the pointers to the left node and right node of it, respectively.

Request: Implement function:

`ETNode* constructET(string s);`

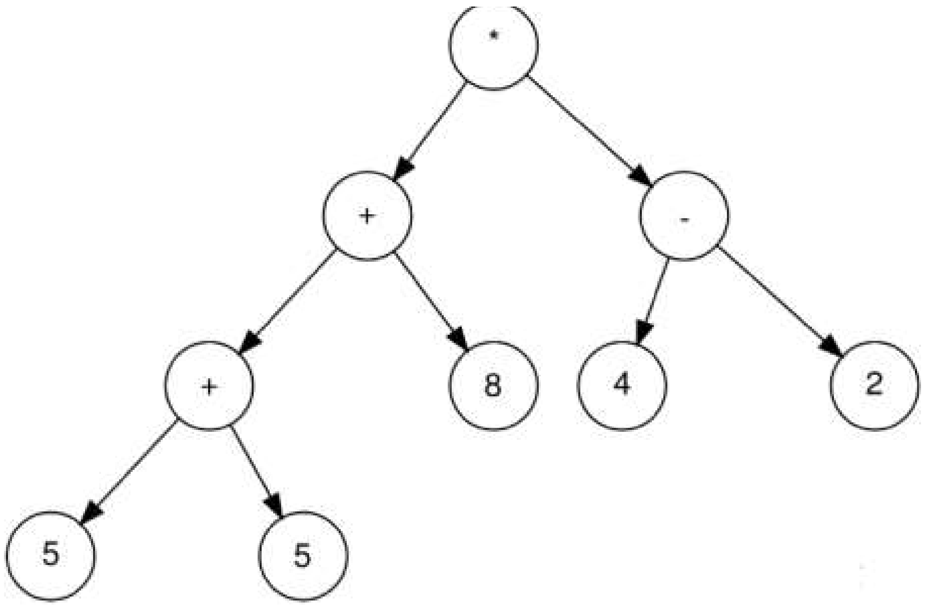
Where `s` is the postorder traversal of an expression tree (the length of string is between 1 and 100000). This function returns the root of the expression tree constructed with the given postorder traversal.

More information:

- You can click here for more information about expression tree.

Example:

The expression tree constructed with postorder `"55+8+42-*"` should be:



Note: In this exercise, the libraries `iostream`, `stack`, `queue`, `vector` and `using namespace std` are used. You can write helper functions; however, you are not allowed to use other libraries.

For example:

Test	Result

`string s = "47*17-3++";`
`ETNode* root = constructET(s);`
`ETNode::printInorder(root);`
`ETNode::deleteTree(root);`
	`4 * 7 + 1 - 7 + 3`

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
ETNode* constructET(string s) {
    stack<ETNode*> stk;

    for (char c : s) {
        if (isdigit(c)) {
            stk.push(new ETNode(c));
            stk.push(new ETNode(c));
        } else {
            ETNode* right = stk.top(); stk.pop();
            ETNode* left = stk.top(); stk.pop();
            stk.push(new ETNode(c, left, right));
        }
    }

    return stk.top();
}
```

	Test	Expected	Got	

✓	`string s = "47*17-3++";`
	`ETNode* root = constructET(s);`
	`ETNode::printInorder(root);`
	`ETNode::deleteTree(root);`
		`4 * 7 + 1 - 7 + 3`	`4 * 7 + 1 - 7 + 3`	✓

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

Câu hỏi 2

Chính xác

Điểm 1,00 của 1,00

Class `ETNode` is used to store a node in expression tree, described on the following:

```
class ETNode {
public:
    char val;
    ETNode *left;
    ETNode *right;
    ETNode() {
        this->left = this->right = NULL;
    };
    ETNode(char val) {
        this->val = val;
        this->left = this->right = nullptr;
    }
    ETNode(char val, ETNode* left, ETNode* right) {
        this->val = val;
        this->left = left;
        this->right = right;
    }
};
```

Where `val` is the value of node (either an operand (`'0'-'9'`) or an operator (`+,-,*`)), `left` and `right` are the pointers to the left node and right node of it, respectively.

Request: Implement function:

`int evaluateET(ETNode* root);`

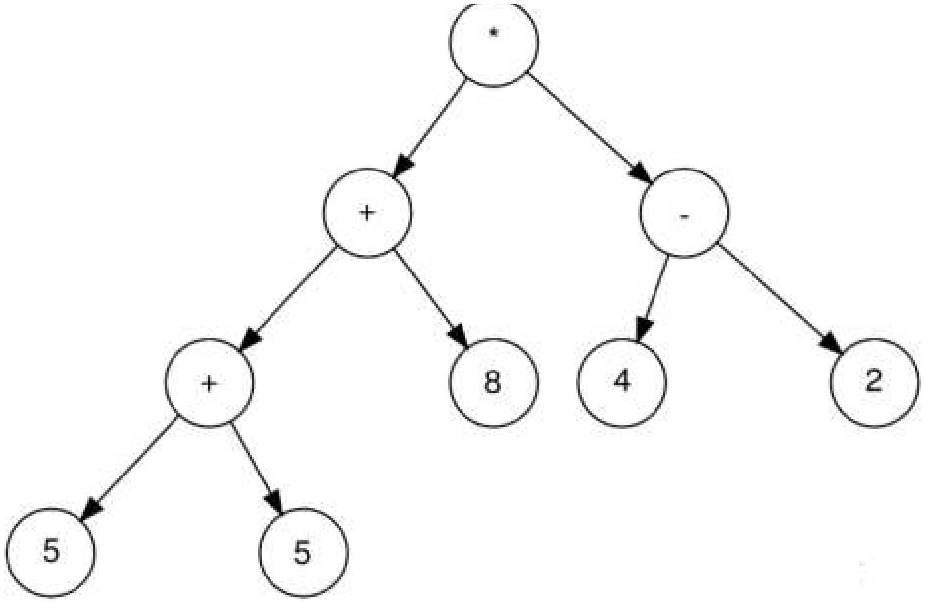
Where `root` is the root node of given expression tree (this tree has between 1 and 100000 elements). This function returns the evaluation of the expression tree. The result may be too large, so the result should mod by `2002`.

More information:

- You can click here for more information about expression tree.

Example:

Given a expression tree in the following:



The result should be `36` (the result of `((5+5)+8)*(4-2)`).

Note: In this exercise, the libraries `iostream`, `stack`, `queue`, `vector` and `using namespace std` are used. You can write helper functions; however, you are not allowed to use other libraries.

For example:

Test	Result

`string s = "47*17-3++";`
`ETNode* root = ETNode::createETWithPostfix(s);`
`cout << evaluateET(root);`
`ETNode::deleteTree(root);`
	25

Answer: (penalty regime: 0, 0, 0, 0, 0, 100 %)

Reset answer

```
int evaluateET(ETNode* root) {
    if (root == nullptr) return 0;

    if (isdigit(root->val)) {
        return root->val - '0';
    } else {
        int leftResult = evaluateET(root->left);
        int rightResult = evaluateET(root->right);

        switch (root->val) {
            case '+':
                return (leftResult + rightResult) % 2002;
            case '-':
                return (leftResult - rightResult) % 2002;
            case '*':
                return (leftResult * rightResult) % 2002;
            default:
                return 0; // Handle invalid operators if needed
        }
    }
}
```

	Test	Expected	Got	

✓	string s = "47*17-3++";
	ETNode* root = ETNode::createETWithPostfix(s);
	cout << evaluateET(root);
	ETNode::deleteTree(root);
		25	25	✓

Passed all tests! ✓

Chính xác

Điểm cho bài nộp này: 1,00/1,00.

BÁCH KHOA E-LEARNING

WEBSITE

HCMUT

MyBK

BKSI

LIÊN HỆ

268 Lý Thường Kiệt, P.14, Q.10, TP.HCM

(028) 38 651 670 - (028) 38 647 256 (Ext: 5258, 5234)

elearning@hcmut.edu.vn