

```
//By Ritwik Chandra Pandey
//On 4th Sep 2021
//Eval postfix - exp trees
```

```
#include<stdio.h>
#include<malloc.h>
struct tree {
    char data;
    struct tree *left;
    struct tree *right;
};
typedef struct tree * ENODE;
ENODE stack[30];
int top = -1;
ENODE newnode(char ch) {
    ENODE temp;
    temp = (ENODE)malloc(sizeof(struct tree));
    temp->data = ch;
    temp->left = NULL;
    temp->right = NULL;
    return(temp);
}
void push(ENODE temp) {
    stack[++top]=temp;
}
ENODE pop() {
    ENODE p;
    p=stack[top--];
    return(p);
}
char eval(ENODE root) {
    if(root == NULL){
        return 0;
    }
    if(root->left==NULL && root->right==NULL){

        return root->data-'0';
    }
}
```

```

    int l_val = eval(root->left);
    int r_val = eval(root->right);
    if(root->data == '+'){
        return l_val + r_val;
    }
    if(root->data == '-'){
        return l_val - r_val;
    }
    if(root->data == '*'){
        return r_val*l_val;
    }
    return l_val/r_val;
}
void main() {
    char postfix_exp[20];
    ENODE temp,t;
    int j,i;
    printf("Enter a postfix expression : ");
    scanf("%s",postfix_exp);
    for(i=0;postfix_exp[i]!='\0';i++) {
        if(postfix_exp[i]=='*' || postfix_exp[i]=='/' || postfix_exp[i]=='+' || postfix_exp[i]=='-') {
            temp=newnode(postfix_exp[i]);
            temp->right=pop();
            temp->left=pop();
            push(temp);
        }
        else {
            temp=newnode(postfix_exp[i]);
            push(temp);
        }
    }
    printf("Result of postfix expression is : ");
    printf("%d\n",eval(temp));
}

```