18. Write a C program to implement the back end of the compiler?

#include <stdio.h>

#include <stdlib.h>

typedef enum { ADD, SUB, MUL, DIV, NUM } NodeType;

typedef struct ASTNode {

NodeType type;

int value; // Only used if type is NUM

struct ASTNode \*left, \*right;

} ASTNode;

ASTNode\* createNumNode(int value) {

ASTNode\* node = (ASTNode\*)malloc(sizeof(ASTNode));

node->type = NUM;

node->value = value;

node->left = node->right = NULL;

return node;

}

ASTNode\* createOpNode(NodeType type, ASTNode\* left, ASTNode\* right) {

ASTNode\* node = (ASTNode\*)malloc(sizeof(ASTNode));

node->type = type;

node->left = left;

node->right = right;

return node;

}

void generateCode(ASTNode\* root) {

if (root == NULL) return;

generateCode(root->left);

generateCode(root->right);

if (root->type == NUM) {

printf("PUSH %d\n", root->value);

} else {

switch (root->type) {

case ADD: printf("ADD\n"); break;

case SUB: printf("SUB\n"); break;

case MUL: printf("MUL\n"); break;

case DIV: printf("DIV\n"); break;

default: break;

}

}

}

int main() {

/\*

Example AST for: (3 + 5) \* (10 - 2)

(\*)

/ \

(+) (-)

/ \ / \

3 5 10 2

\*/

ASTNode\* root = createOpNode(MUL,

createOpNode(ADD, createNumNode(3), createNumNode(5)),

createOpNode(SUB, createNumNode(10), createNumNode(2))

);

printf("Generated Assembly Code:\n");

generateCode(root);

return 0;

}

