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

**Program:**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_SYMBOLS 100

#define MAX\_CODE\_LENGTH 1000

typedef struct {

char name[50];

int value;

} Symbol;

typedef struct {

char code[MAX\_CODE\_LENGTH];

int lineNumber;

} IntermediateCode;

Symbol symbolTable[MAX\_SYMBOLS];

IntermediateCode intermediateCode[MAX\_CODE\_LENGTH];

int symbolCount = 0;

int codeCount = 0;

void addSymbol(char \*name, int value) {

strcpy(symbolTable[symbolCount].name, name);

symbolTable[symbolCount].value = value;

symbolCount++;

}

void generateCode(char \*code) {

strcpy(intermediateCode[codeCount].code, code);

intermediateCode[codeCount].lineNumber = codeCount + 1;

codeCount++;

}

void printSymbolTable() {

printf("Symbol Table:\n");

for (int i = 0; i < symbolCount; i++) {

printf("Name: %s, Value: %d\n", symbolTable[i].name, symbolTable[i].value);

}

}

void printIntermediateCode() {

printf("Intermediate Code:\n");

for (int i = 0; i < codeCount; i++) {

printf("%d: %s\n", intermediateCode[i].lineNumber, intermediateCode[i].code);

}

}

int main() {

addSymbol("x", 10);

addSymbol("y", 20);

generateCode("LOAD x");

generateCode("ADD y");

generateCode("STORE z");

printSymbolTable();

printIntermediateCode();

return 0;

}

**Output:**

