Name: Atandrit Chatterjee

Reg. No.: RA2011031010042

**Experiment-12** **Simple Code Generator**

Aim:

To implement simple code generator in C/C++.

Procedure:

1. Parse the input code and generate an abstract syntax tree (AST).
2. Traverse the AST and generate intermediate code (e.g. three-address code).
3. Optimize the intermediate code to improve performance.
4. Generate target code (e.g. machine code) from the optimized intermediate code.
5. Output the generated code to a file or execute it directly.

Code:

#include <iostream>

#include <cstdio>

#include <cstring>

#include <cctype>

using namespace std;

typedef struct

{

char var[10];

int alive;

} regist;

regist preg[10];

void substring(char exp[], int st, int end)

{

int i, j = 0;

char dup[10] = "";

for (i = st; i < end; i++)

dup[j++] = exp[i];

dup[j] = '0';

strcpy(exp, dup);

}

int getregister(char var[])

{

int i;

for (i = 0; i < 10; i++)

{

if (preg[i].alive == 0)

{

strcpy(preg[i].var, var);

break;

}

}

return (i);

}

void getvar(char exp[], char v[])

{

int i, j = 0;

char var[10] = "";

for (i = 0; exp[i] != '\0'; i++)

if (isalpha(exp[i]))

var[j++] = exp[i];

else

break;

strcpy(v, var);

}

int main()

{

char basic[10][10], var[10][10], fstr[10], op;

int i, j, k, reg, vc, flag = 0;

cout << "\nEnter the Three Address Code:\n";

for (i = 0;; i++)

{

cin.getline(basic[i], 10);

if (strcmp(basic[i], "exit") == 0)

break;

}

cout << "\nThe Equivalent Assembly Code is:\n";

for (j = 0; j < i; j++)

{

getvar(basic[j], var[vc++]);

strcpy(fstr, var[vc - 1]);

substring(basic[j], strlen(var[vc - 1]) + 1, strlen(basic[j]));

getvar(basic[j], var[vc++]);

reg = getregister(var[vc - 1]);

if (preg[reg].alive == 0)

{

printf("\nMov R%d,%s", reg, var[vc - 1]);

preg[reg].alive = 1;

}

op = basic[j][strlen(var[vc - 1])];

substring(basic[j], strlen(var[vc - 1]) + 1, strlen(basic[j]));

getvar(basic[j], var[vc++]);

switch (op)

{

case '+':

cout << "\nAdd";

break;

case '-':

cout << "\nSub";

break;

case '\*':

cout << "\nMul";

break;

case '/':

cout << "\nDiv";

break;

}

flag = 1;

for (k = 0; k <= reg; k++)

{

if (strcmp(preg[k].var, var[vc - 1]) == 0)

{

cout << "R" << k << ", R" << reg;

preg[k].alive = 0;

flag = 0;

break;

}

}

if (flag)

{

printf(" %s,R%d", var[vc - 1], reg);

printf("\nMov %s,R%d", fstr, reg);

}

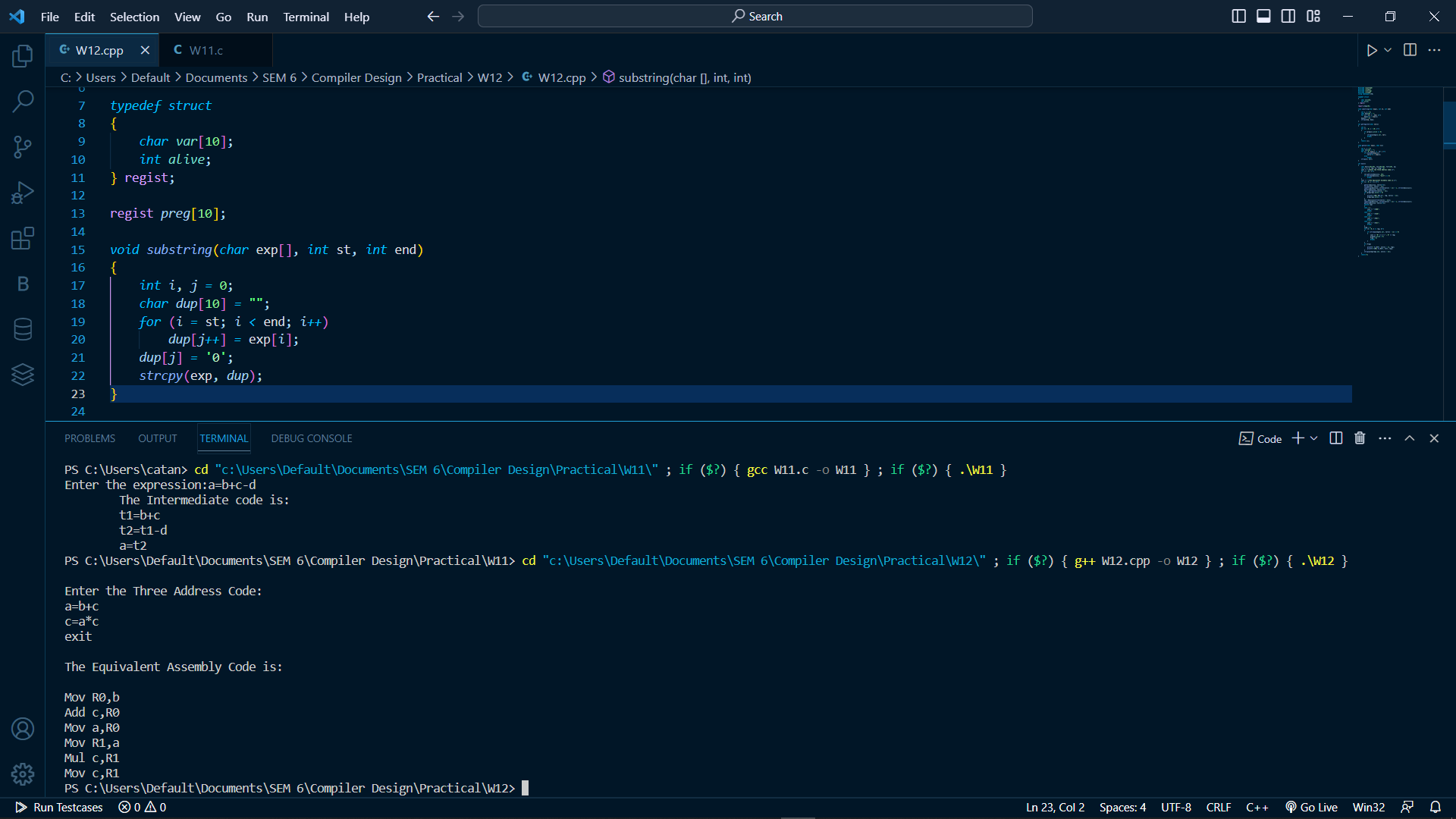
strcpy(preg[reg].var, var[vc - 3]);

}

return 0;

}

Output:



Result:

The implementation of simple code generator was successful.