#include <cstring>

#include <iostream>

#include <vector>

#include <numeric>

#include <sstream>

#include <iterator>

#include <sstream>

using namespace std;

int x;

char \* save2[100];

char \* save3[100];

int splitting(char str[100], char \* delim) {

vector < string > str1;

char \* copy = strdup(str);

/

char \* res = strtok(str, delim);

int count = 0;

char dm[50];

while (res) {

str1.push\_back(res);

int from = res - str + strlen(res);

res = strtok(NULL, delim);

int to = res != NULL ? res - str : strlen(copy);

sprintf(dm, "%.\*s", to - from, copy + from);

str1.push\_back(dm);

}

std::ostringstream vts;

if (!str1.empty()) {

std::copy(str1.begin(), str1.end() - 1,

std::ostream\_iterator < string > (vts, ","));

vts << str1.back();

}

string from\_ostr = vts.str();

const char \* coch\_save = from\_ostr.c\_str();

char \* save = strdup(coch\_save);

char \* delim1 = ",";

char \* res1 = strtok(save, delim1);

int count2 = 0;

while (res1) {

save2[count2] = res1;

res1 = strtok(NULL, delim1);

count2++;

}

return count2;

}

int splitting2(char str[100], char \* delim) {

vector < string > str1;

char \* copy = strdup(str);

char \* res = strtok(str, delim);

int count = 0;

char dm[50];

while (res) {

str1.push\_back(res);

int from = res - str + strlen(res);

res = strtok(NULL, delim);

int to = res != NULL ? res - str : strlen(copy);

sprintf(dm, "%.\*s", to - from, copy + from);

str1.push\_back(dm);

}

std::ostringstream vts;

if (!str1.empty()) {

std::copy(str1.begin(), str1.end() - 1,

std::ostream\_iterator < string > (vts, ","));

vts << str1.back();

}

string from\_ostr = vts.str();

const char \* coch\_save = from\_ostr.c\_str();

char \* save = strdup(coch\_save);

char \* delim1 = ",";

char \* res1 = strtok(save, delim1);

int count2 = 0;

while (res1) {

save3[count2] = res1;

res1 = strtok(NULL, delim1);

count2++;

}

return count2;

}

int check\_simpleno(char st[50]) {

int j = 0, state = 0;

int table[5][3] = {

{

1,

4,

4

},

{

1,

2,

4

},

{

3,

4,

4

},

{

3,

4,

4

},

{

4,

4,

4

}

};

int input;

int flag = 0;

while (st[j] != '\0') {

if (st[j] == '.') {

input = 1;

} else if (isdigit(st[j])) {

input = 0;

} else {

input = 2;

}

state = table[state][input];

if (state == 1) {

flag = 1;

} else if (state == 3) {

flag = 1;

} else {

flag = 0;

}

j++;

}

if (flag == 1) {

return 1;

} else {

return 0;

}

}

int check\_identifier(char st[50]) {

int j = 0, state = 0;

int table[3][4] = {

{

1,

1,

2,

2

},

{

1,

1,

1,

2

},

{

2,

2,

2,

2

}

};

int input;

int flag = 0;

while (st[j] != '\0') {

if (st[j] == '\_') {

input = 0;

} else if (isalpha(st[j])) {

input = 1;

} else if (isdigit(st[j])) {

input = 2;

} else {

input = 3;

}

state = table[state][input];

if (state == 1) {

flag = 1;

} else {

flag = 0;

}

j++;

}

if (flag == 1) {

return 1;

} else {

return 0;

}

}

int check\_expression(char st[50]) {

char \* str\_exp[100];

int j = 0, state = 0;

int l;

int table[5][4] = {

{

1,

4,

1,

4

},

{

4,

2,

4,

4

},

{

3,

4,

3,

4

},

{

4,

2,

4,

4

},

{

4,

4,

4,

4

}

};

int input;

int flag = 0;

l = splitting2(st, "+/-\*%");

for (int j = 0; j < l; j++) {

str\_exp[j] = save3[j];

if (check\_identifier(str\_exp[j]) == 1) {

input = 0;

} else if (strcmp(str\_exp[j], "+") == 0 || strcmp(str\_exp[j], "-") == 0 || strcmp(str\_exp[j], "/") == 0 || strcmp(str\_exp[j], "\*") == 0 || strcmp(str\_exp[j], "%") == 0) {

input = 1;

} else if (check\_simpleno(str\_exp[j]) == 1) {

input = 2;

} else{ input = 3;

}

state = table[state][input];

if (state == 3) {

flag = 1;

} else {

flag = 0;

}

}

if (flag == 1) {

return 1;

} else {

return 0;

}

}

int main() {

char str[100];

int l;

char \* str\_va[100];

int state = 0;

int table[7][7] = {

{

1,

6,

6,

6,

6,

6,

6

},

{

6,

3,

6,

5,

6,

6,

2

},

{

6,

3,

6,

6,

6,

6,

6

},

{

1,

6,

4,

6,

6,

4,

6

},

{

6,

6,

6,

5,

6,

6,

6

},

{

6,

6,

6,

6,

6,

6,

6

},

{

6,

6,

6,

6,

6,

6,

6

}

};

int input;

int flag = 0;

cin >> str;

l = splitting(str, "=;");

int h;

for (int j = 0; j < l; j++) {

str\_va[j] = save2[j];

if (check\_identifier(str\_va[j]) == 1) {

input = 0;

} else if (strcmp(str\_va[j], "=") == 0) {

input = 1;

} else if (check\_expression(str\_va[j]) == 1) {

input = 2;

} else if (strcmp(str\_va[j], ";") == 0) {

input = 3;

} else if (strcmp(str\_va[j], "+") == 1 || strcmp(str\_va[j], "-") == 1 || strcmp(str\_va[j], "/") == 1 || strcmp(str\_va[j], "\*") == 1 || strcmp(str\_va[j], "%") == 1) {

input = 6;

}else if (check\_simpleno(str\_va[j])==0)

{

input=5;

}

else { input = 4;}

state = table[state][input];

if (state == 5) {

flag = 1;

}

}

if (flag == 1) {

cout << "VALID ASSIGNMENT STATEMENT";

} else {

cout << "INVALID ASSIGNMENT STATEMENT";

}

return 0;

}