**PRINCIPLES OF COMPILER DESIGN**

**20XW64**

**-20PW39**

#include<iostream>

#include<string>

using namespace std;

int fail(int s) {

if(s>=0 && s<=46) {

return 47;

}

else if(s>=47 && s<=49) {

return 50;

}

else if(s>=50 && s<=52) {

return 53;

}

else if(s>=53 && s<=58) {

return 59;

}

else if(s>=59 && s<=61) {

return 62;

}

}

void lexicalAnalyze(string str) {

int n = str.size();

int s = 0, i = 0;

while (true) {

switch(s) {

case 0:

if (str[i] == 'i') {

s = 1;

i++;

}

else if (str[i] == 'c') {

s = 19;

i++;

}

else if (str[i] == 'm') {

s = 33;

i++;

}

else if (str[i] == 's') {

s = 38;

i++;

}

else {

s = fail(s);

i = 0;

}

break;

case 1:

if (str[i] == 'n') {

s = 2;

i++;

}

else if (str[i] == 'o') {

s = 11;

i++;

}

else

{

s = fail(s);

i=0;

}

break;

case 2:

if (str[i] == 'c'){

s = 3;

i++;

}

else if(str[i] == 't'){

s = 9;

i++;

}

else {

s = fail(s);

i = 0;

}

break;

case 3:

if (str[i] == 'l') {

s = 4;

i++;

}

else {

s=fail(s);

i=0;

}

break;

case 4:

if (str[i] == 'u') {

s = 5;

i++;

}

else {

s=fail(s);

i=0;

}

break;

case 5:

if (str[i] == 'd') {

s = 6;

i++;

}

else {

s=fail(s);

i=0;

}

break;

case 6:

if (str[i] == 'e') {

s = 7;

i++;

}

else {

s=fail(s);

i=0;

}

break;

case 7:

if (str[i] == '\0') {

s = 8;

i++;

}

else {

s=fail(s);

i=0;

}

break;

case 8:

cout << "It is an keyword" << endl;

return;

case 9:

if (str[i] == '\0') {

s=10;

i++;

} else {

s = fail(s);

i=0;

}

break;

case 10:

cout << "It is a keyword" << endl;

return;

case 11:

if (str[i] == 's') {

s = 12;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 12:

if (str[i] == 't') {

s = 13;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 13:

if (str[i] == 'r') {

s = 14;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 14:

if (str[i] == 'e') {

s = 15;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 15:

if (str[i] == 'a') {

s = 16;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 16:

if (str[i] == 'm') {

s = 17;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 17:

if (str[i] == '\0') {

s = 18;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 18:

cout << "It is a keyword" << endl;

return;

case 19:

if (str[i] == 'a') {

s = 20;

i++;

}

else if (str[i] == 'o') {

s = 26;

i++;

}

else if (str[i] == 'i') {

s = 30;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 20:

if (str[i] == 'l') {

s = 21;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 21:

if (str[i] == 'l') {

s = 22;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 22:

if (str[i] == 'o') {

s = 23;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 23:

if (str[i] == 'c') {

s = 24;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 24:

if (str[i] == '\0') {

s = 25;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 25:

cout << "It is a keyword" << endl;

return;

case 26:

if (str[i] == 'u') {

s = 27;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 27:

if (str[i] == 't') {

s = 28;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 28:

if (str[i] == '\0') {

s = 29;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 29:

cout << "It is a keyword" << endl;

return;

case 30:

if (str[i] == 'n') {

s = 31;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 31:

if (str[i] == '\0') {

s = 32;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 32:

cout << "It is a keyword" << endl;

return;

case 33:

if (str[i] == 'a') {

s = 34;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 34:

if (str[i] == 'i') {

s = 35;

i++;

}

else if (str[i] == 'l'){

s = 21;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 35:

if (str[i] == 'n') {

s = 36;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 36:

if (str[i] == '\0') {

s = 37;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 37:

cout << "It is a keyword" << endl;

return;

case 38:

if (str[i] == 't') {

s = 39;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 39:

if (str[i] == 'd') {

s = 40;

i++;

}

else if (str[i]=='r') {

s = 42;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 40:

if (str[i] == '\0') {

s = 41;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 41:

cout << "It is a keyword" << endl;

return;

case 42:

if (str[i] == 'i') {

s = 43;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 43:

if (str[i] == 'n') {

s = 44;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 44:

if (str[i] == 'g') {

s = 45;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 45:

if (str[i] == '\0') {

s = 46;

i++;

}

else {

s = fail(s);

i=0;

}

break;

case 46:

cout << "It is a keyword" << endl;

return;

case 47:

if (isalpha(str[i]) || str[i]=='\_') {

s = 48;

i++;

}

else {

s = fail(s);

i = 0;

}

break;

case 48:

if (isalpha(str[i]) || isdigit(str[i]) || str[i]=='\_') {

i++;

} else if (str[i] == '\0') {

s = 49;

i++;

}

else {

s = fail(s);

i = 0;

}

break;

case 49:

cout << "It is an identifier" << endl;

return;

case 50:

if (isdigit(str[i])) {

s = 51;

i++;

}

else {

s = fail(s);

i = 0;

}

break;

case 51:

if (isdigit(str[i])) {

i++;

}

else if (str[i] == '\0') {

s = 52;

i++;

}

else {

s = fail(s);

i = 0;

}

break;

case 52:

cout << "It is a constant" << endl;

return;

case 53:

if (str[i] == '<') {

s = 54;

i++;

}

else if (str[i] == '>') {

s = 54;

i++;

}

else if (str[i] == '=') {

s = 54;

i++;

}

else if (str[i] == '!') {

s = 54;

i++;

}

else if (str[i] == '&') {

s = 55;

i++;

}

else if (str[i] == '|') {

s = 56;

i++;

}

else if (str[i] == '+') {

s = 54;

i++;

}

else if (str[i] == '-') {

s = 54;

i++;

}

else if (str[i] == '\*') {

s = 54;

i++;

}

else if (str[i] == '/') {

s = 54;

i++;

}

else if (str[i] == '%') {

s = 54;

i++;

}

else {

s = fail(s);

i = 0;

}

break;

case 54:

if (str[i] == '\0') {

s = 58;

i++;

}

else if (str[i] == '=') {

s = 57;

i++;

}

else

{

s = fail(s);

i = 0;

}

break;

case 55:

if (str[i] == '&') {

s = 57;

i++;

}

else if(str[i] == '\0') {

s = 58;

i++;

}

else

{

s = fail(s);

i = 0;

}

break;

case 56:

if (str[i] == '|') {

s = 57;

i++;

}

else if(str[i] == '\0') {

s = 58;

i++;

}

else

{

s = fail(s);

i = 0;

}

break;

case 57:

if (str[i]=='\0') {

s = 58;

i++;

}

else

{

s = fail(s);

}

break;

case 58:

cout << "It is an operator" << endl;

return;

case 59:

if (str[i] = '\'' || str[i] == ')' || str[i] == '(' || str[i] == '/' || str[i] == '}' || str[i] == '{' || str[i] == '(' || str[i] == ')' || str[i] == ';' || str[i] == '"') {

s = 60;

i++;

}

else

{

s = fail(s);

i = 0;

}

break;

case 60:

if (str[i] == '\0') {

s = 61;

i++;

}

else

{

s = fail(s);

i=0;

}

break;

case 61:

cout << "It is a delimiter" << endl;

return;

case 62:

cout << "It is not a token" << endl;

return;

}

}

}

int main() {

string str;

while(true) {

cout << "Enter the string: ";

cin >> str;

lexicalAnalyze(str);

cout << endl;

getchar();

}

}