

Compiler Design Lab

Soham Bhardwaj, 2K19/IT/122

Program-1

Objective: Write a program in C to identify a token by using the keywords if or switch and verify it.

Code:

```
#include <stdbool.h>
#include <stdio.h>
#include <string.h>
#include <ctype.h>

bool isKeyword(char* str) {
    char keywords[][10] =
{"auto", "break", "case", "char", "const", "continue", "default",
"do", "double", "else", "enum", "extern", "float", "for", "goto",
"if", "int", "long", "register", "return", "short", "signed",
"sizeof", "static", "struct", "switch", "typedef", "union",
"unsigned", "void", "volatile", "while", "main", "class"};
    for(int i = 0; i < (int)sizeof(keywords)/(int)sizeof(keywords[0]); ++i){
        if(strcmp(keywords[i], str) == 0) return true;
    }
    return false;
}

bool isDelimiter(char* str) {
    if(strlen(str)>1) return false; // for checking identifier
    char delim[]=".,;(){}";
    for(int i=0;i<strlen(delim);i++) {
        if(delim[i]==str[0]) return true;
    }
    return false;
}

bool isOperator(char* str) {
    char operator[][3]={"+", "-", "*", "/", "=", "++", "+=", "-=", "*=", "/=",
"--", "%", "<", ">", "<=", ">=", "!=", "==",
"&&", "||", ">>", "<<", "&", "?", ":"};
```

```

    for(int i=0;i<(int)sizeof(operator)/(int)sizeof(operator[0]);i++) {
        if(strcmp(str,operator[i])==0) return true;
    }
    return false;
}

bool isIdentifier(char* str) {
    if( isDelimiter(str) || isOperator(str) || isKeyword(str) || (str[0]!='_' &&
!isalpha(str[0])) ) return false;
    for(int i=1;i<strlen(str);i++) {
        if(!isdigit(str[i]) && !isalpha(str[i]) && !(str[i]=='_')) return false;
    }
    return true;
}

bool isNumber(char* str) {
    if(strlen(str)==1) return isdigit(str[0]);
    if (str[0] != '.' && str[0] != '+' && str[0] != '-' && !isdigit(str[0])) return
false;
    int flagDot=0, flagE=0;
    for (int i=0; i <strlen(str); i++) {
        if (str[i] != 'e' && str[i] != '.' && str[i] != '+' && str[i] != '-' &&
!isdigit(str[i]) ) return false;
        if (str[i] == '.') {
            flagDot++;
            if(flagE>0) return false;
            if( i+1<strlen(str) && !isdigit(str[i+1])) return false;
            if (flagDot >1) return false;
        }
        else if (str[i] == 'e') {
            flagE++;
            if(flagE>1) return false;
            if (!isdigit(str[i-1])) return false;
            if (i + 1 > strlen(str)) return false;
            if (str[i + 1] != '+' && str[i + 1] != '-' && !isdigit(str[i+1])) return
false;
        }
    }
    return true;
}

void check(char* token) {
    int len=strlen(token);
    if(isKeyword(token)) printf("Keyword\n");
    else if (len<3 && isOperator(token)) printf("Operator\n");
    else if (len==1 && isDelimiter(token)) printf("Seperator/Delimiter\n");
}

```

```

    else if (isIdentifier(token)) printf("Identifier\n");
    else if (isNumber(token)) printf("Number\n");
    else printf("Invalid Token\n");
}
int main() {
    char token[200];
    printf("Enter token: ");
    scanf("%s", token);
    check(token);
}

```

Examples :

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```

soham@SBMacBook Desktop % ./program
Enter token: for
Keyword
soham@SBMacBook Desktop % ./program
Enter token: _JAM12
Identifier
soham@SBMacBook Desktop % ./program
Enter token: 12e12
Number
soham@SBMacBook Desktop % ./program
Enter token: -.1234
Number
soham@SBMacBook Desktop % ./program
Enter token: 12.1e-10
Number
soham@SBMacBook Desktop % ./program
Enter token: >=
Operator
soham@SBMacBook Desktop % ./program
Enter token: ++
Operator
soham@SBMacBook Desktop % ./program
Enter token: }
Seperator/Delimiter
soham@SBMacBook Desktop % ./program
Enter token: ;
Seperator/Delimiter
soham@SBMacBook Desktop % ./program
Enter token: S_123
Identifier
soham@SBMacBook Desktop % ./program
Enter token: 124_S
Invalid Token
soham@SBMacBook Desktop % ./program
Enter token: ab9.
Invalid Token
soham@SBMacBook Desktop % █

```