

## **PROJECT NO : 6**

**AIM:** Write an appropriate language description for a layman language which can do mathematical operations using English like sentences.

**EXAMPLE:**

Add 100 ,200,300,400 . Sub 250 from result.

Mul 400 to it . Div the answer by 2. Show me the answer.

**Lab3: Scanner phase implementation of assigned project in "C" language.**

**Group Members:**

1. Kathiriya Darshak (IT057)
2. Limbani Nihal (IT064)
3. Karia Stuti (IT055)

We use "c++" instead of "c".

Regular Defination	Examples:
Keywords	From, from, Show, show, by, to, it, result, the, me, Answer, answer
Operation	Add, Sub(Subtract), Mul(Multiply), Div(Divide)
Digit(Number)	[0-9]
Que. Mark	“?”(EOF)
White Space	(Tab   Newline) <sup>+</sup>
Letter	[A-Za-z]

## Code:

```
#include <bits/stdc++.h>

using namespace std;

int operation(char buf[])
{
    char op[10][10]={"Add","add","Sub","sub","Mul","mul","Div","div"};
    for(int i=0;i<8;i++)
    {
        if(strcmp(op[i],buf)==0)
            return 1;
    }
    return 0;
}

int keyword(char buf[])
{
    char key[20][20]={"From", "from", "Show", "show", "by", "to", "it", "number", "and", "result", "Result",
    "the", "me", "Answer", "answer" };
    for(int i=0;i<15;i++)
    {
        if(strcmp(key[i],buf)==0)
            return 1;
    }
    return 0;
}

int number(char buf[])
{
    int n=strlen(buf);
    for(int i=0;i<n;i++)
    {
        if(buf[i]>='0' && buf[i]<='9')
        {
            return 1;
        }
    }
}
```

```

        else
        {
            return 0;
        }
    }
    return 1;
}

int main()
{

    FILE *f;

    f=fopen("D:/00 Study/SEM 6/OLAB/LT/LAB 3/inputExpfile2.txt", "r", stdin);
    freopen("D:/00 Study/SEM 6/OLAB/LT/LAB 3/output2.txt", "w", stdout);

    char ch,buffer[15];

    int d=0;

    while((ch = fgetc(f)) != EOF){
        if(ch=='\n')
        {
            printf("Started New Line. \n");

            continue;
        }
        if(ch=='?')
        {
            printf("It shown end of line: %c\n",ch);

            continue;
        }
        if(isalnum(ch))
        {
            buffer[d++]=ch;
        }

        else if((ch==' ' || ch=='\n') && (d!=0)){
            buffer[d]='\0';

            d=0;

```

```

if(ch!='')
{
    if(number(buffer)==1){
        printf("Number identify: %s\n", buffer);
    }
}
else{
    printf("\n");
    continue;
}
if(operation(buffer)==1)
{
    printf("operator identify: %s\n",buffer);
}
else if(keyword(buffer)==1)
{
    printf("Keyword identify: %s\n",buffer);
}
}
else if(ch==' ' || ch==' ')
{
    continue;
}
else
{
    printf("Not identify token %c\n",ch);
}
}
}

```

**Input File:**

```
inputExpfile - Notepad
File Edit Format View Help
Add 100 ,200 ,300 ,400 ?
Sub 250 from result ?
Mul 400 to it ?
Div the answer by 2 ?
Show me the answer ?
```

**Output File:**

```
output - Notepad
File Edit Format View Help
operator identify: Add
Number identify: 100
Number identify: 200
Number identify: 300
Number identify: 400
It shown end of line: ?
Started New Line.
operator identify: Sub
Number identify: 250
Keyword identify: from
Keyword identify: result
It shown end of line: ?
Started New Line.
operator identify: Mul
Number identify: 400
Keyword identify: to
Keyword identify: it
It shown end of line: ?
Started New Line.
operator identify: Div
Keyword identify: the
Keyword identify: answer
Keyword identify: by
Number identify: 2
It shown end of line: ?
Started New Line.
Keyword identify: Show
Keyword identify: me
Keyword identify: the
Keyword identify: answer
It shown end of line: ?
Started New Line.
```

### Input File2: Show case sensitive

inputExpfile2 - Notepad

File Edit Format View Help

```
add 100 ,200 ,300 ,400 ?  
sub 250 From Result ?  
mul 400 to it ?  
div the Answer by 2 ?  
show me the Answer ?
```

### Output File2:

output2 - Notepad

File Edit Format View Help

```
operator identify: add  
Number identify: 100  
Number identify: 200  
Number identify: 300  
Number identify: 400  
It shown end of line: ?  
Started New Line.  
operator identify: sub  
Number identify: 250  
Keyword identify: From  
Keyword identify: Result  
It shown end of line: ?  
Started New Line.  
operator identify: mul  
Number identify: 400  
Keyword identify: to  
Keyword identify: it  
It shown end of line: ?  
Started New Line.  
operator identify: div  
Keyword identify: the  
Keyword identify: Answer  
Keyword identify: by  
Number identify: 2  
It shown end of line: ?  
Started New Line.  
Keyword identify: show  
Keyword identify: me  
Keyword identify: the  
Keyword identify: Answer  
It shown end of line: ?  
Started New Line.
```