**Virtual Lab Experiment**

**Design Lexical Analyzer in C/C++ -**

#include<bits/stdc++.h>

using namespace std;

string keywords[]={"int","float","if","else","while","for"};

string operators[]={"<",">","<=",">=","==","=","+","-","++","--

"};

char punctuation[]={'(',')',',',';','{','}','[',']'};

vector<string> k,o,c,i;

vector<string> p;

/\*

This function checks whether the given word is a keyword in

the list of keywords present.

If it is a keyword, it is added into the vector 'k'.

\*/

void search\_for\_keywords(string a ,int flag)

{

if(a.size()==0)return;

int size=sizeof(keywords)/sizeof(keywords[0]);

for(int i=0;i<size;i++)

{

if(a==keywords[i])

{

k.push\_back(a);

return;

}

}

if(!flag) i.push\_back(a);

}

/\*

This function checks if the given character is present in

the given array of punctuation marks.

If it is present, it is added into the vector 'p'.

\*/

bool search\_for\_punctuation(char a )

{

//cout<<a;

int size=sizeof(punctuation)/sizeof(punctuation[0]);

for(int i=0;i<size;i++)

{

//cout<<punctuation[i];

if(a==punctuation[i])

{

//cout<<a;

string temp=" ";

temp[0]=a;

2

//cout<<temp;

p.push\_back(temp);

return true;

}

}

return false;

}

/\*

This function is to create the tokens of integers and

floatingpoint integers.

\*/

void search\_for\_constants(string a )

{

if(a.size()>0) c.push\_back(a);

}

/\*

This function prints the list/vector of strings and the

number of strings which is provided as the input.

\*/

void print(vector<string>a )

{

cout<<"\n";//cout<<"---------------------------------------

-\n";

for(int i=0;i<a.size();i++)

{

cout<<a[i]<<" ";

}

cout<<"\nTotal="<<a.size()<<"\n";

cout<<"----------------------------\n";

}

/\*

This function checks if the given input is a part of the

list of operators defined.

If it is a part of the list, it is added into the vector of

operators 'o'.

\*/

void search\_for\_operators(string line,int& i)

{

// This is to check the operators which are composed of two

characters like '++', '+='.

string temp=line.substr(i,2);

//cout<<temp<<endl;

int size=sizeof(operators)/sizeof(operators[0]);

for(int j=0;j<size;j++)

{

//cout<<punctuation[i];

if(temp==operators[j])

{

o.push\_back(temp);i=i+1;

3

return;

}

}

// This is to check the operators which are composed of

only one character like '+', '-'.

temp=line.substr(i,1);

//cout<<temp<<endl;

for(int j=0;j<size;j++)

{

//cout<<punctuation[i];

if(temp==operators[j])

{

o.push\_back(temp);

return;

}

}

}

int main()

{

cout<<"Enter number of lines of input:";

int n;

cin>>n;n++;

while(n--)

{

char arr[100];

cin.getline(arr,100,'\n');

string line=arr;

//cout<<line<<line.length();

int i;

string cur="";

string cur\_num="";

int flag=0,flag2=0;

int no\_of\_dots=0;

for(i=0;i<line.length();i++)

{

char now=line[i];

if((now>='a'&&now<='z')||(now>='A'&&now<='Z'))

{ // Check for keywords and identifiers starts.

if(now=='e'&&flag==0&&no\_of\_dots>0){cur\_num+=now;continue;}

if(cur\_num.size()>0){flag2=1;}

flag=1;

cur+=line[i]; }

else if(now==' '){

// Found a delimiter, hence checking the stored

input till now for keywords and constants.

if(flag)search\_for\_keywords(cur,flag2);

4

else if(!flag2) search\_for\_constants(cur\_num);

cur="";flag=0;cur\_num="";no\_of\_dots=0; }

else if(now>='0'&&now<='9'||now=='.')

{ //Check for number starts. Keeping count of

number of '.'s.

if(now=='.'&&no\_of\_dots>0)cur\_num="";

else if(now=='.')no\_of\_dots++;

if(flag)cur+=line[i];

else cur\_num+=line[i];

}

else{

//cout<<now;

// If none of the above conditions pass, this

block of code checks for all of the keywords, numbers, operators

and punctuations.

if((now=='+'||now=='-

')&&flag==0&&cur\_num.size()>0){cur\_num+=now;continue;}

if(flag)search\_for\_keywords(cur,flag2);

else if(!flag2) search\_for\_constants(cur\_num);

cur="";flag=0;cur\_num="";no\_of\_dots=0;

if(!search\_for\_punctuation(now))

search\_for\_operators(line,i);

;//cout<<now;}}

//If still some are not matched, search for keywords and

numbers.

if(flag)search\_for\_keywords(cur,flag2);

else if(!flag2) search\_for\_constants(cur\_num);

cur="";flag=0;cur\_num="";}

cout<<"\n\nKeywords:";

print(k);cout<<"Operators:";

print(o);cout<<"Constants:";

print(c);cout<<"Punctation:";

print(p);cout<<"Identifiers:";

print(::i);

cout<<"Total tokens

are:"<<k.size()+o.size()+c.size()+p.size()+::i.size()<<"\n";

return 0;

}

**Output -**



