Lab Report

Compiler Design

Name: Marjuk Ahmed Siddiki

ID: 171-15-8959

Sec: J

Daffodil International University

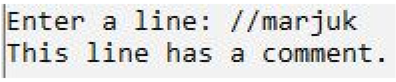
13th December, 2019

Check the given line comment or not

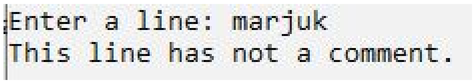
Code:

```
#include<iostream>
#include<string>
#include<vector>
using namespace std;
int main()
{
   string s;
   vector <string> v_s;
   int c=0;
   cout<<"Enter a line: ";
   cin>>s;
   v_s.push_back(s);
   for(int i=0;i<v_s[0].size();i++){
      if(v_s[0].at(i)=='/' && v_s[0].at(i+1)=='/'){
         cout<<"This line has a comment."<<endl;
         c++;
         break;
      }
   }
   if(c==0){
      cout<<"This line has not a comment."<<endl;
   }
   return 0;
}
```

Output



```
Enter a line: //marjuk
This line has a comment.
```



```
Enter a line: marjuk
This line has not a comment.
```

← END →

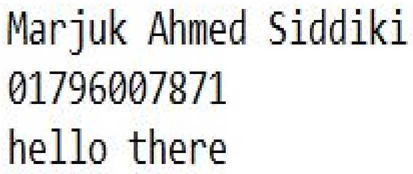
Word separation from a paragraph

Code:

```
#include<iostream>
#include<string>
#include<fstream>
using namespace std;
int main()
{
    ifstream file;
    file.open("paragraph.txt");
    string word;
    int c=0;
    while(file >> word){
        cout<<word<<endl;
        c++;
    }
    cout<<endl<<"Total word: "<<c<<endl;
    c=0;
    file.close();
    return 0;
}
```

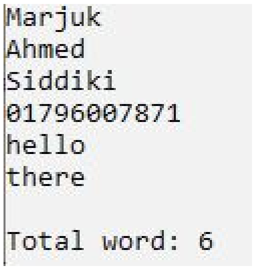
Output

Input File -> some sentence



```
Marjuk Ahmed Siddiki
01796007871
hello there
```

Output -> consol



```
Marjuk
Ahmed
Siddiki
01796007871
hello
there

Total word: 6
```

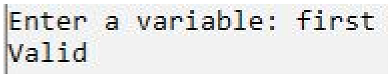
← END →

Variable check (valid or invalid)

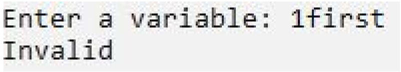
Code:

```
#include<iostream>
#include<string>
#include<vector>

using namespace std;

int main()
{
    string s;
    vector <string> v_s;
    int c=0;

    cout<<"Enter a variable: ";
    cin>>s;
    v_s.push_back(s);

    if(v_s[0].at(0)>='a' && v_s[0].at(0)<='z' || v_s[0].at(0)>='A' && v_s[0].at(0)<='Z' ||
v_s[0].at(0)=='_' || v_s[0].at(0)=='$'){
        c++;
    }

    for(int i=1;i<v_s[0].size();i++){
        if(v_s[0].at(i)>='a' && v_s[0].at(i)<='z' || v_s[0].at(i)>='A' && v_s[0].at(i)<='Z' ||
v_s[0].at(i)>='0' && v_s[0].at(i)<='9' || v_s[0].at(i)=='_' || v_s[0].at(i)=='$'){
            c++;
        }
    }
    if(c==v_s[0].size()){
        cout<<"Valid"<<endl;
    }else{
        cout<<"Invalid"<<endl;
    }
    return 0;
}
```

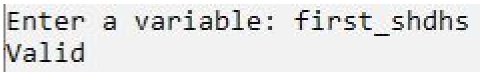
Output



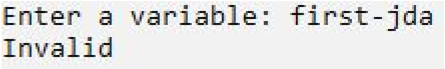
```
Enter a variable: first
Valid
```



```
Enter a variable: 1first
Invalid
```



```
Enter a variable: first_shdhs
Valid
```



```
Enter a variable: first-jda
Invalid
```

← END →

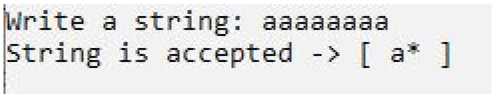
Recognized string under "a*","a*b+", "abb"

Code:

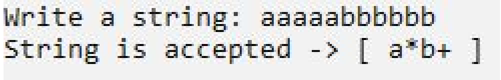
```
#include<iostream>
#include<string>
#include<vector>

using namespace std;

int main()
{
    string s;
    vector <string> v_s;
    int a=0,c=0;

    cout<<"Write a string: ";
    cin>>s;

    v_s.push_back(s);

    for(int i=0;i<v_s[0].size();i++){
        if(v_s[0].at(i)=='a' || v_s[0].at(0)==' '){
            a++;
            if(i==v_s[0].size()-1){
                break;
            }
        }
        if(v_s[0].at(i)=='a' && v_s[0].at(i+1)=='b' || v_s[0].at(0)==' ' && v_s[0].at(i+1)=='b'){
            cout<<"String is accepted -> [ a*b+ ]"<<endl;
            c=1;
            break;
        }
        if(s.at(v_s[0].size()-1)=='b' && v_s[0].at(v_s[0].size()-2)=='b' &&
v_s[0].at(v_s[0].size()-3)=='a'){
```

```
            cout<<"String is accepted -> [ abb ]"<<endl;
            c=1;
            break;
        }
    }

    if(a==v_s[0].size()){
        cout<<"String is accepted -> [ a* ]"<<endl;
    }else if(c==0){
        cout<<"String is not accepted."<<endl;
    }

    return 0;
}
```

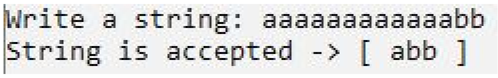
Output



```
Write a string: aaaaaaaa
String is accepted -> [ a* ]
```



```
Write a string: aaaaabbbbbb
String is accepted -> [ a*b+ ]
```



```
Write a string: aaaaaaaaaaaabb
String is accepted -> [ abb ]
```

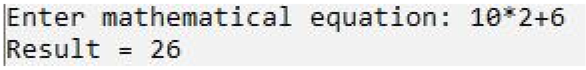
← END →

Mathematical Expression Evaluation

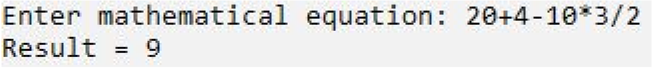
Code:

```
#include<iostream>
#include<string>
#include<vector>

using namespace std;

int main()
{
    string equ,load,s;
    char c;
    int sl;
    vector <int> vector_n;
    vector <char> vector_o;

    cout<<"Enter mathematical equation: ";
    cin>>equ;

    for(int i=0;i<=equ.size();i++){
        if(equ[i]>='0' && equ[i]<='9'){
            s = equ[i];
            load = load.append(s);
        }else{

            sl = stoi(load);
            vector_n.push_back(sl);
            load = "";
        }
        if(equ[i]=='+' || equ[i]=='-' || equ[i]=='*' || equ[i]=='/'){
            c = equ[i];
            vector_o.push_back(c);
    }
}

int num1,num2,save,i=0;
vector <char> oprt;

oprt.push_back('*');
oprt.push_back('/');
oprt.push_back('+');
oprt.push_back('-');

start:
    if(vector_o.empty()){
        goto endd;
    }else{
        while(i<oprt.size()){
            for(int l=0;l<vector_o.size();l++){
                if(oprt[i] == vector_o[l]){
                    num1 = vector_n[l];
                    num2 = vector_n[l+1];

                    if(oprt[i] == '*'){
                        save = num1 * num2;
                    }else if(oprt[i] == '/'){
                        save = num1 / num2;
                    }else if(oprt[i] == '+'){
                        save = num1 + num2;
                    }else if(oprt[i] == '-'){
                        save = num1 - num2;
                    }

                    vector_n[l] = save;

                    if(vector_n[l+1]==vector_n[vector_n.size()-1]){
                        vector_n.pop_back();
                        }else{
                            for(int j=l+1;j<vector_n.size()-1;j++){
                                vector_n[j] = vector_n[j+1];
                            }
                            vector_n.pop_back();
                        }
                        if(vector_o[l+1]=='+' || vector_o[l+1]=='-' || vector_o[l+1]=='*' ||
vector_o[l+1]=='/'){
                            for(int j=l;j<vector_o.size()-1;j++){
                                vector_o[j] = vector_o[j+1];
                            }
                            vector_o.pop_back();
                        }else{
                            vector_o.pop_back();
                        }
                        l--;
                    }
                }
                i++;
            }
            i=0;
            goto start;
        }
        endd:

    cout<<"Result = "<<vector_n[0]<<endl;



    return 0;
}
```

Output



```
Enter mathematical equation: 10*2+6
Result = 26
```



```
Enter mathematical equation: 20+4-10*3/2
Result = 9
```

← *END* →

First and Follow

Code:

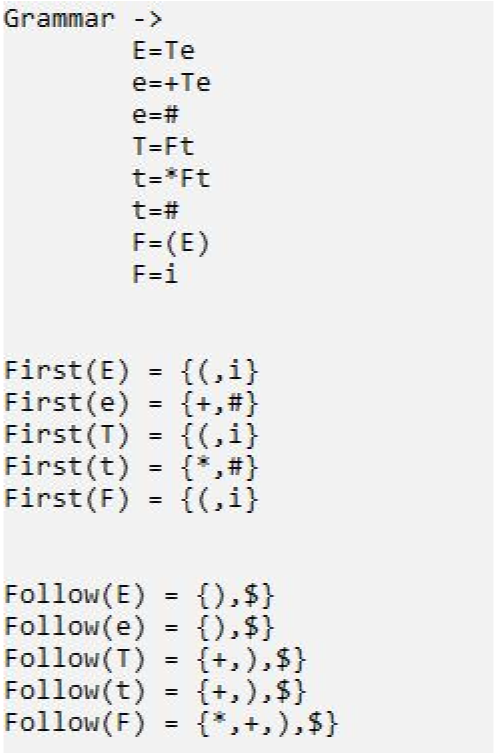
```
#include<iostream>
#include<string>
#include<vector>

using namespace std;

int main() {
   vector <string> grammer;

   grammer.push_back("E=Te");
   grammer.push_back("e=+Te");
   grammer.push_back("e=#");
   grammer.push_back("T=Ft");
   grammer.push_back("t=*Ft");
   grammer.push_back("t=#");
   grammer.push_back("F=(E)");
   grammer.push_back("F=i");

   cout<<"Grammar ->"<<endl;

   for(int i=0; i!=grammer.size(); i++) {
      cout<<"\t"<<grammer[i]<<"\n";
   }
   cout<<"\n\n";

   int i=0;
start_first:
   while(i!=grammer.size()) {
      if((i>0)&&(grammer[i].at(0)==grammer[i-1].at(0))) {
         i++;
```

```
        continue;
    } else {
        cout<<"First("<<grammer[i].at(0)<<") = {";
        for(int j=0; j!=grammer.size(); j++) {
            if(grammer[i].at(2)==grammer[j].at(0)) {
                for(int k=0; k!=grammer.size(); k++) {
                    if(grammer[j].at(2)==grammer[k].at(0)) {
                        cout<<grammer[k].at(2)<<",";
                        if(grammer[k].at(0)==grammer[k+1].at(0)) {
                            cout<<grammer[k+1].at(2);
                        }
                        cout<<"}\n";
                        i++;
                        goto start_first;
                    }
                }
                cout<<grammer[j].at(2)<<",";
                if(grammer[j].at(0)==grammer[j+1].at(0)) {
                    cout<<grammer[j+1].at(2);
                }
                cout<<"}\n";
                i++;
                goto start_first;
            }
        }
        cout<<grammer[i].at(2)<<",";
        if(grammer[i].at(0)==grammer[i+1].at(0)) {
            cout<<grammer[i+1].at(2);
        }
        cout<<"}\n";
        i++;
        goto start_first;
    }
}
cout<<endl<<endl;
```

```
    string save,rr,last=",$}";
    vector<string> v_s;
    int a=0,e=0;
start_follow:
    while(a!=grammer.size()) {
        if((a>0)&&(grammer[a].at(0)==grammer[a-1].at(0))) {
            a++;
            continue;
        } else {
            rr = grammer[a].at(0);
            save.append(rr);
            save.append("=");
            cout<<"Follow("<<grammer[a].at(0)<<") = {";
            for(int b=0; b!=grammer.size(); b++) {
                for(int c=2; c!=grammer[b].size(); c++) {
                    if(grammer[a].at(0)==grammer[b].at(c)) {
                        if(grammer[b].at(c)==grammer[b].at(grammer[b].size()-1)) {
                            for(int e=0; e!=grammer.size(); e++) {
                                if(grammer[b].at(0)==v_s[e].at(0)) {
                                    for(int f=2; f!=v_s[e].size(); f++) {
                                        cout<<v_s[e].at(f);
                                        rr = v_s[e].at(f);
                                        save.append(rr);
                                    }
                                    cout<<endl;
                                    a++;
                                    v_s.push_back(save);
                                    save = "";
                                    goto start_follow;
                                }
                            }
                        } else {
                            for(int d=0; d!=grammer.size(); d++) {
                                if(grammer[b].at(c+1)==grammer[d].at(0)) {
                                    cout<<grammer[d].at(2)<<",";
        rr = grammer[d].at(2);
        save.append(rr);
        rr = ",";
        save.append(rr);
        if(grammer[d].at(0)==grammer[d+1].at(0)) {
            if(grammer[d+1].at(2)=='#') {
                for(int f=2; f!=v_s[b].size(); f++) {
                    cout<<v_s[b].at(f);
                    rr = v_s[b].at(f);
                    save.append(rr);
                }
                cout<<endl;
                a++;
                v_s.push_back(save);
                save = "";
                goto start_follow;
            } else {
                cout<<grammer[d+1].at(2);
                rr = grammer[d+1].at(2);
                save.append(rr);
            }
        }
        cout<<endl;
        a++;
        v_s.push_back(save);
        save = "";
        goto start_follow;

    }
}
rr = grammer[b].at(c+1);
save.append(rr);
cout<<grammer[b].at(c+1)<<",$}"<<endl;
save.append(last);
v_s.push_back(save);
```

```
                    save = "";

                    a++;
                    goto start_follow;
                  }
                }
              }
            }
          }
        }

        return 0;
      }
```

Output



```
Grammar ->
        E=Te
        e=+Te
        e=#
        T=Ft
        t=*Ft
        t=#
        F=(E)
        F=i


First(E) = {(,i}
First(e) = {+,#}
First(T) = {(,i}
First(t) = {*,#}
First(F) = {(,i}


Follow(E) = {),$}
Follow(e) = {),$}
Follow(T) = {+,),$}
Follow(t) = {+,),$}
Follow(F) = {*,+,),$}
```

← *END* →

Source code: http://bit.ly/compilerM