

Name: Rachit Shah

Roll No: 20BCE266

Course Name and Course Code: 2CS701 Compiler Construction

Practical No: 6

Aim: Intermediate Code Generation: To generate Three Address code for assignment statement.

Source Code:

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

using namespace std;
int temp=1;
vector<string>ans;

map<char,int>ops;
void compute(string s){
    cout<<"RHS "<<s<<endl;
    while(1){
        string p="";
        for(int i=0;i<s.size();i++){
            if(s[i]=='/'){
                int j=i-1,k=i+1;
                string lhs="";
                string rhs="";
                while(j>=0 && !ops.count(s[j])){
                    lhs+=s[j];
                    j--;
                }
                while(k<s.size() && !ops.count(s[k])){
                    rhs+=s[k];
                    k++;
                }
                reverse(lhs.begin(),lhs.end());
                string a = to_string(temp);
                ans.push_back("t" + a + " = " + lhs + " / " + rhs);
                p = s.substr(0,j+1) + "t" + a + s.substr(k,s.size() - k + 1);
                s=p;
                break;
            }
        }
        if(p=="")break;
        temp++;
    }
    while(1){
```

```

string p="";
for(int i=0;i<s.size();i++){
    if(s[i]=='*'){
        int j=i-1,k=i+1;string lhs="";
        string rhs="";
        while(j>=0 && !ops.count(s[j])){
            lhs+=s[j];
            j--;
        }
        while(k<s.size() && !ops.count(s[k])){
            rhs+=s[k];
            k++;
        }
        reverse(lhs.begin(),lhs.end());
        string a = to_string(temp);
        ans.push_back("t" + a + " = " + lhs + " * " + rhs);
        p = s.substr(0,j+1) + "t" + a + s.substr(k,s.size() - k + 1);
        s=p;
        break;
    }
}
if(p=="")break;
temp++;
}
while(1){
    string p="";
    for(int i=0;i<s.size();i++){
        if(s[i]=='+'){
            int j=i-1,k=i+1;string lhs="";
            string rhs="";
            while(j>=0 && !ops.count(s[j])){
                lhs+=s[j];
                j--;
            }
            while(k<s.size() && !ops.count(s[k])){
                rhs+=s[k];
                k++;
            }
            reverse(lhs.begin(),lhs.end());
            string a = to_string(temp);
            ans.push_back("t" + a + " = " + lhs + " + " + rhs);
            p = s.substr(0,j+1) + "t" + a + s.substr(k,s.size() - k + 1);
            s=p;
            break;
        }
    }
    if(p=="")break;
    temp++;
}

```

```

    }
    while(1){
        string p="";
        for(int i=0;i<s.size();i++){
            if(s[i]=='-'){
                int j=i-1,k=i+1;string lhs="";
                string rhs="";
                while(j>=0 && !ops.count(s[j])){
                    lhs+=s[j];
                    j--;
                }
                reverse(lhs.begin(),lhs.end());
                while(k<s.size() && !ops.count(s[k])){
                    rhs+=s[k];
                    k++;
                }
                string a = to_string(temp);
                ans.push_back("t" + a + " = " + lhs + " - " + rhs);
                p = s.substr(0,j+1) + "t" + a + s.substr(k,s.size() - k + 1);
                s=p;
                break;
            }
        }
        if(p=="")break;
        temp++;
    }
}

string preprocess(string s){
    string r="";
    stack<char>st;
    for(int i=0;i<s.size();i++){
        if(s[i]!='('){
            st.push(s[i]);
        }
        else{
            string lhs="";
            string rhs="";
            int f=0;
            char p;
            while(st.top()!='('){
                if(ops.count(st.top())){
                    p=st.top();
                    f=1;
                }
                else if(f==0)rhs+=st.top();
                else lhs+=st.top();
                st.pop();
            }
        }
    }
}

```

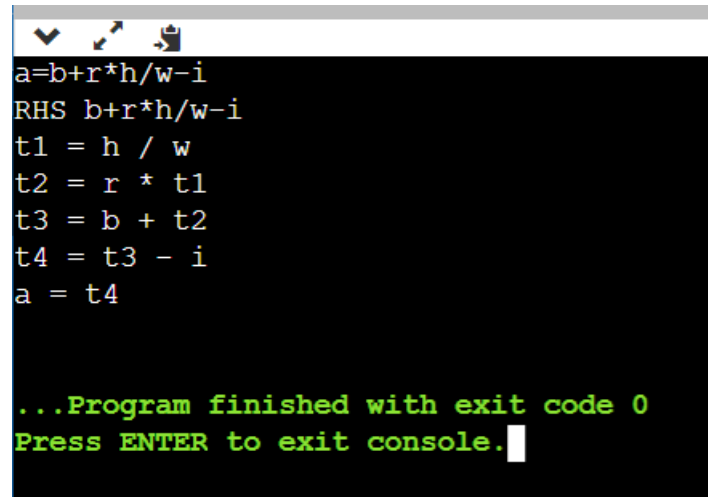
```

        st.pop();
        reverse(lhs.begin(),lhs.end());
        reverse(rhs.begin(),rhs.end());
        string a = to_string(temp);
        ans.push_back("t" + a + " = " + lhs + " " + p + " " + rhs);
        st.push('t');
        temp++;
        for(int j=0;j<a.size();j++)st.push(a[j]);
    }
}
while(st.size()>0){
    r+=st.top();
    st.pop();
}
reverse(r.begin(),r.end());
return r;
}
int main()
{
    string s;
    cin>>s;
    ops['-'] = 1;
    ops['+'] = 1;
    ops['*'] = 1;
    ops['/'] = 1;
    int f=-1;
    s=preprocess(s);
    char p;
    for(int i=0;i<s.size();i++){
        if(s[i]=='='){
            string rhs = s.substr(i+1,s.size()-i-1);
            compute(rhs);
            string a = to_string(temp-1);
            if(f==-1){
                ans.push_back(string(1,s[0]) + " = t" + a);
            }
            else{
                ans.push_back(string(1,s[0]) + " = " + string(1,s[0]) + " " +
string(1,p) + " t" + a);
            }
            break;
        }
    }
    for(auto c:ops){
        if(s[i]==c.first){
            f=1;
            p=c.first;
            break;
        }
    }
}

```

```
    }  
    }  
}  
for(auto c:ans)cout<<c<<endl;  
}
```

Output:



```
a=b+r*h/w-i  
RHS b+r*h/w-i  
t1 = h / w  
t2 = r * t1  
t3 = b + t2  
t4 = t3 - i  
a = t4  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```