

Program-9

Q) Write a program to generate three address code of an expression.

Code:-

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

using namespace std;

inline void debugMode() {
    #ifndef ONLINE_JUDGE
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif
}

bool isLetter(char ch) {
    return ((ch>='A' && ch<='Z') || (ch>='a' && ch<='z'));
}

bool isDigit(char ch) {
    return (ch>='0' && ch<='9');
}

bool isOperator(char ch) {
    return (ch=='+' || ch=='-' || ch=='*' || ch=='/');
}

int main() {
    debugMode();

    string str;
    cin>>str;

    stack<string> st1;
    stack<char> st2;
    vector<string> vec;

    int n = str.size();

    int cnt=1;

    string s;
```

```
vector<string> ans;
string finalOut;
for(int i=0; i<n; i++) {
    if(str[i] == ' ') continue;

    if(isLetter(str[i])) {
        if(vec.size() == 0) {
            string s1;
            s1.push_back(str[i]);
            vec.push_back(s1);
        }
        else if(vec.size() == 2) {
            string s1;
            s1.push_back(str[i]);
            vec.push_back(s1);

            string temp;
            string tempVar;

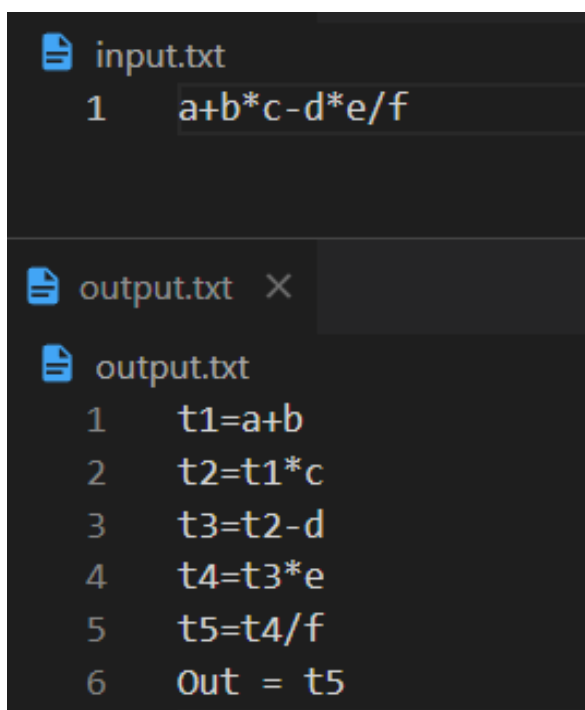
            tempVar.push_back('t');
            string tempVarNum = to_string(cnt);
            cnt++;
            for(auto it : tempVarNum) {
                tempVar.push_back(it);
            }
            finalOut = tempVar;
            for(auto it : tempVar) {
                temp.push_back(it);
            }
            temp.push_back('=');
            for(auto it : vec) {
                for(auto it1:it)
                    temp.push_back(it1);
            }
        }
    }
}
```

```

        vec.clear();
        ans.push_back(temp);
        vec.push_back(tempVar);
        st1.push(tempVar);
    }
}
else if(isOperator(str[i])) {
    if(vec.size() == 1) {
        string s1;
        s1.push_back(str[i]);
        vec.push_back(s1);
    }
}
}
for(auto it : ans) {
    cout<<it<<"\n";
}
cout<<"Out = "<<finalOut<<"\n";
return 0;
}

```

Output:-



The screenshot shows a code editor with two files open: `input.txt` and `output.txt`. The `input.txt` file contains a single line of input: `1 a+b*c-d*e/f`. The `output.txt` file contains the output of the program, which is a sequence of intermediate calculations followed by the final result: `1 t1=a+b`, `2 t2=t1*c`, `3 t3=t2-d`, `4 t4=t3*e`, `5 t5=t4/f`, and `6 Out = t5`.

```

input.txt
1 a+b*c-d*e/f

output.txt
1 t1=a+b
2 t2=t1*c
3 t3=t2-d
4 t4=t3*e
5 t5=t4/f
6 Out = t5

```

Program-10

Q) Write a program to check whether a string belongs to given grammar or not.

Code:-

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

using namespace std;

inline void debugMode() {
    #ifndef ONLINE_JUDGE
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
    #endif
}

int main() {
    debugMode();
    // S->aS
    // S->Sb
    // S->ab
    string str;
    cin>>str;
    int fl=0;
    int cnt1=0, cnt2=0;
    for(auto it : str) {
        if(it == 'a') {
            if(cnt2 != 0) {
                fl=1;
                break;
            }
            cnt1++;
        }
        else if(it == 'b') {
            if(cnt1 == 0) {
```

```

        fl=1;
        break;
    }
    cnt2++;
}
else {
    fl=1;
    break;
}
}
if(fl) {
    cout<<"Given string doesn't satisfy the grammer\n";
}
else {
    cout<<"Given string satisfies the grammer\n";
}
return 0;
}

```

Output:-

```

input.txt
1 aaabbbbbbbb

output.txt ×
output.txt
1 Given string satisfies the grammer

```

```

input.txt
1 aaabbbbbbaa

output.txt ×
output.txt
1 Given string doesn't satisfy the grammer

```