Experiment Number 13

Aim: Implementation of DAG.

Algorithm:

Step 1: Start

Step 2: Read basic blocks of code.

Step 3: Identify operations and operands.

Step 4: Create DAG.

Step 5: Print respective output.

Step 6: Stop.

Code:

```
#include<iostream>
#include<string>
#include<unordered_map>
using namespace std;
class DAG
{   public:
    char label;
    char data;
    DAG* left;
    DAG* right;

    DAG(char x){
      label='_';
      data=x;
      left=NULL;
      right=NULL;
    }
    DAG(char lb, char x, DAG* lt, DAG* rt){
      label=lb;
      data=x;
      left=lt;
      right=rt;
    }
};

int main(){
    int n;cout<<"Enter number of basic blocks: ";
    cin>>n;
    string st[n];
    for(int i=0;i<n;i++)
    cin>>st[i];
    unordered_map<char, DAG*> labelDAGNode;
```

```
    for(int i=0;i<n;i++){
        string stTemp=st[i];
        for(int j=0;j<5;j++){
            char tempLabel = stTemp[0];
            char tempLeft = stTemp[2];
            char tempData = stTemp[3];
            char tempRight = stTemp[4];
            DAG* leftPtr;
            DAG* rightPtr;
            if(labelDAGNode.count(tempLeft) == 0){
                leftPtr = new DAG(tempLeft);
            }
            else{
                leftPtr = labelDAGNode[tempLeft];
            }
            if(labelDAGNode.count(tempRight) == 0){
                rightPtr = new DAG(tempRight);
            }
            else{
                rightPtr = labelDAGNode[tempRight];
            }
            DAG* nn = new DAG(tempLabel,tempData,leftPtr,rightPtr);
            labelDAGNode.insert(make_pair(tempLabel,nn));
        }
    }
    cout<<"Label      ptr/op      leftPtr       rightPtr"<<endl;
    for(int i=0;i<n;i++){
        DAG* x=labelDAGNode[st[i][0]];
        cout<<st[i][0]<<"           "<<x->data<<"           ";
        if(x->left->label=='_')cout<<x->left->data;
        else cout<<x->left->label;
        cout<<"          ";
        if(x->right->label=='_')cout<<x->right->data;
        else cout<<x->right->label;
        cout<<endl;
    }
    return 0;
}
```

Output:

```
Enter number of basic blocks: 4
a=b+c
c=b*d
f=b-e
s=c+f
Label      ptr/op      leftPtr       rightPtr
a            +            b          c
c            *            b          d
f            -            b          e
s            +            c          f
```

Result: Thus, Implementation of DAG done successfully.