Experiment Number 14

Aim: Implementation of Global Data Flow Analysis.

Algorithm:

Step 1: Start

Step 2: Read code.

Step 3: Analyse expression by expression.

Step 4: Create logically in order data flow statements.

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 isin(string a,char b){
    for(int i=0;i<a.length();i++)
    if(a[i]==b)
    return 1;
    return 0;
}
int main(){
    int n;cout<<"Enter number of basic blocks: ";
```

```
cin>>n;
string st[n];
string vars="";
for(int i=0;i<n;i++)
{cin>>st[i];
for(int x=0;x<5;x++){
   if(isalpha(st[i][x]))
   if(isin(vars,st[i][x])==0)
   vars+=st[i][x];
}}
cout<<"Variables identified: "<<vars<<endl;
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<<"SNo     Label     ptr/op     leftPtr     rightPtr"<<endl;
for(int i=0;i<n;i++){
   DAG* x=labelDAGNode[st[i][0]];
   cout<<(i+1)<<"          "<<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;
}
for(int z=0;z<vars.length();z++){
   int init=0;
```

```
        cout<<"Variable: "<<vars[z]<<endl;
        for(int i=0;i<n;i++)
            for(int j=0;j<5;j++)
            if(st[i][j]==vars[z] and j==0){
                if(init==0){
                cout<<"Initialized at ["<<i+1<<"]"<<endl;
                init=i+1;}
            else{
                cout<<"Updated at ["<<i+1<<"]"<<endl;
                init=i+1;}
        }
        else if(st[i][j]==vars[z] and j!=0)
        cout<<"Used at ["<<i+1<<"]"<<endl;
    }
    return 0;
}
```

Output:

```
Enter number of basic blocks: 3
a=a+1
b=a*c
c=d/f
Variables identified: abcdf
SNo      Label      ptr/op      leftPtr       rightPtr
1            a           +           a          1
2            b           *           a          c
3            c           /           d          f
Variable: a
Initialized at [1]
Used at [1]
Used at [2]
Variable: b
Initialized at [2]
Variable: c
Used at [2]
Initialized at [3]
Variable: d
Used at [3]
Variable: f
Used at [3]
```

Result: Thus, Implementation of Global Data Flow Analysis done successfully.