ROLL NO:46

NAME: Saurabh Khandagale

Session: EVEN 2020-2021

Compiler Design Lab

PRACTICAL No. 6

Topic: Code Optimization

Platform: Windows or Linux

Language to be used: Python or Java (based on the companies targeted for placement)

Aim: Write a program to perform loop detection by finding leader, basic blocks and program flow graph.

Code:

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package codeoptimization;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.InputStreamReader;
import java.util.*;
import java.util.Map;
/**
 *
 * @author ADMIN
 */

class numberComparator implements Comparator<String>
{



   public int compare(String s1,String s2)
   {
      String st1[]=s1.split(" ");
      String st2[]=s2.split(" ");
      int d1=Integer.parseInt(st1[0].replace(")",""));
```

```
        int d2=Integer.parseInt(st2[0].replace(")",""));

        if(d1==d2)
        {
           return 0;
        }

        else if(d1>d2)
        {
           return 1;
        }

        else
        {
           return -1;
        }

     }
}


public class CodeOPtimization {

    public static void main(String[] args)throws Exception
        {
           BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
//           System.out.println("Enter the number of Statements:");
//           int t=Integer.parseInt(br.readLine());
//           System.out.println("Enter the "+t+" statements 1 by 1");
           ArrayList leader=new ArrayList();
           ArrayList statements=new ArrayList();
           int t=0;
            System.out.println("The file contains the following data:");
           try
           {
              File myObj = new File("input.txt");
              Scanner myReader = new Scanner(myObj);
              while (myReader.hasNextLine()) {
                String data = myReader.nextLine();
                 System.out.println(data);
                 statements.add(data);
                 t++;
              }
              myReader.close();
           }
          catch (FileNotFoundException e)
           {
            System.out.println("An error occurred.");
            e.printStackTrace();
           }
```

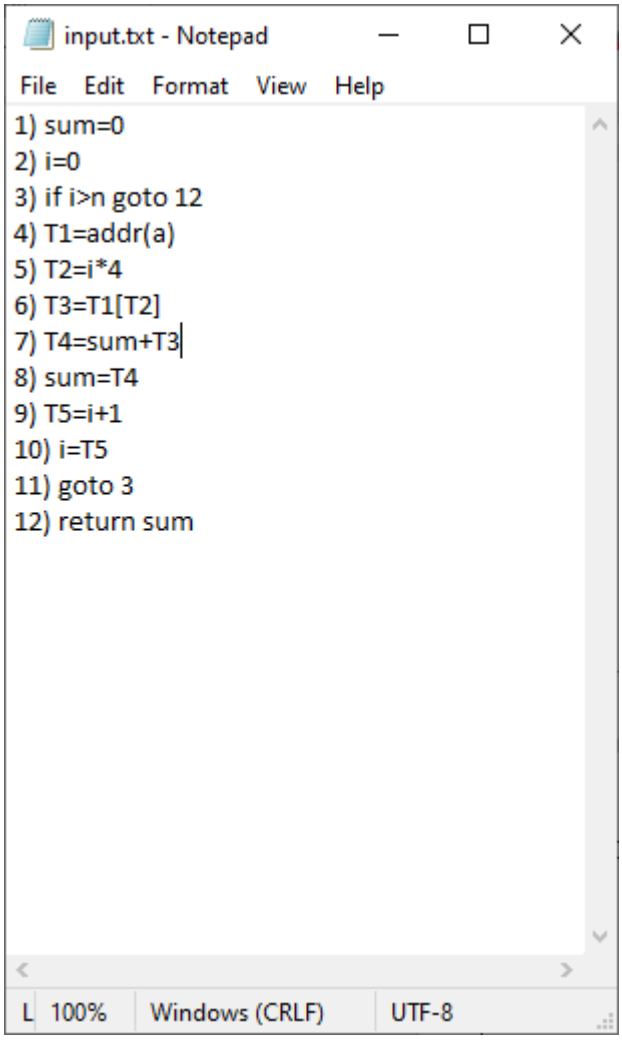
```
for(int i=0;i<t;i++)
{
   String str=statements.get(i)+"";


   if(i==0)
   {
      leader.add(str);
      continue;
   }
   if(str.contains("goto") && !str.contains("if"))
   {
      String st[]=str.split(" ");
      int in=Integer.parseInt(st[2]);
      String str1=statements.get(in-1)+"";
      leader.add(str1);
      continue;
   }
   if(str.contains("goto") && str.contains("if"))
   {
      String st[]=str.split(" ");
      int in=Integer.parseInt(st[4]);
      String str1=statements.get(in-1)+"";
      leader.add(str1);
      continue;
   }

   String prev=statements.get(i-1)+"";


   if(prev.contains("if"))
   {
      leader.add(str);
      continue;
   }



}
//System.out.println(leader);
Collections.sort(leader,new numberComparator());
System.out.println("---------------------------------------------- ");
System.out.println("LEADERS ARE: ");
for(Object l: leader)
```

```
{
   System.out.println(l+"");
}
System.out.println("---------------------------------------------------- ");
//System.out.println(statements);
Map<Integer, List<String>> blocks = new HashMap<Integer, List<String>>();
//HashMap<Integer,String> blocks=new HashMap<Integer,String>();
int k=1;

 // System.out.println(blocks);
for(int i=0;i<statements.size();)
{
  String s=statements.get(i)+"";

  if(leader.contains(s))
  {
     int start=statements.indexOf(s);
     int lindex=leader.indexOf(s);
      String nextleader;
      int nextindex,end;
     if(lindex!=leader.size()-1)
     {
         nextleader=leader.get(lindex+1)+"";
         nextindex=statements.indexOf(nextleader);
         end=nextindex-1;
     }
     else
     {
        end=statements.size()-1;
     }

     ArrayList list=new ArrayList();

     while(i<=end)
     {
        list.add(statements.get(i));
        i++;
     }

     blocks.put(k,list);
     k++;

  }


}
System.out.println("---------------------------------------------------- ");
System.out.println("BLOCKS ARE: ");
for(int i=0;i<blocks.size();i++)
{
   System.out.println("Block B"+(i+1)+" : "+blocks.get(i+1));
```

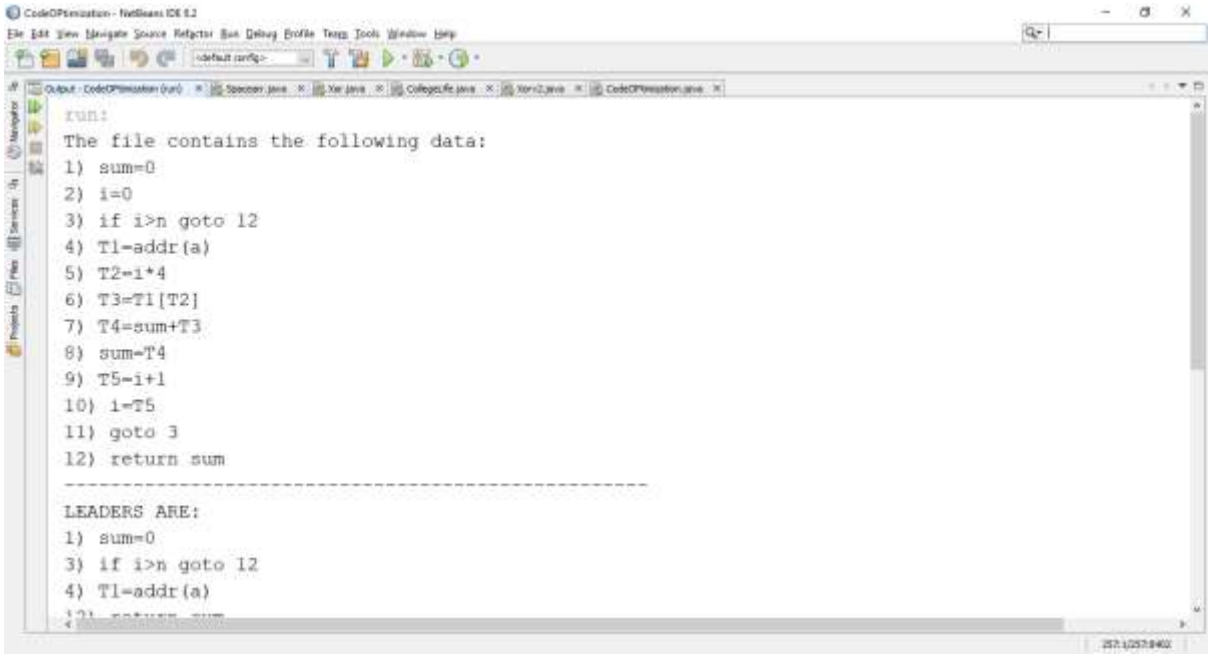
```
}
System.out.println("---------------------------------------------------- ");


System.out.println("---------------------------------------------------- ");
System.out.println("THE FLOW IS:");
for(int i=0;i<blocks.size();i++)
{
   List list=new ArrayList();
   list=blocks.get(i+1);
   if(i==blocks.size()-1)
   {
      System.out.print("Block B"+(i+1)+" -> "+"END");
   }
   else
   {
      System.out.print("Block B"+(i+1)+" -> "+" B"+(i+2));
   }


   for(int j=0;j<list.size();j++)
   {
      String str=list.get(j)+"";

      if(str.contains("goto") && !str.contains("if"))
      {
         String st[]=str.split(" ");
         String in=st[2];
         for(int z=0; z<blocks.size();z++)
         {
            list=blocks.get(z+1);
             for(int m=0;m<list.size();m++)
             {
                if((list.get(m)+"").contains(in))
                {
                   if((z+1)!=(i+1))
                   {
                     System.out.print(" B"+(z+1));
                   }

                }
             }
         }

         continue;
      }

      if(str.contains("goto") && str.contains("if"))
      {
        String st[]=str.split(" ");
         String in=st[4];
```

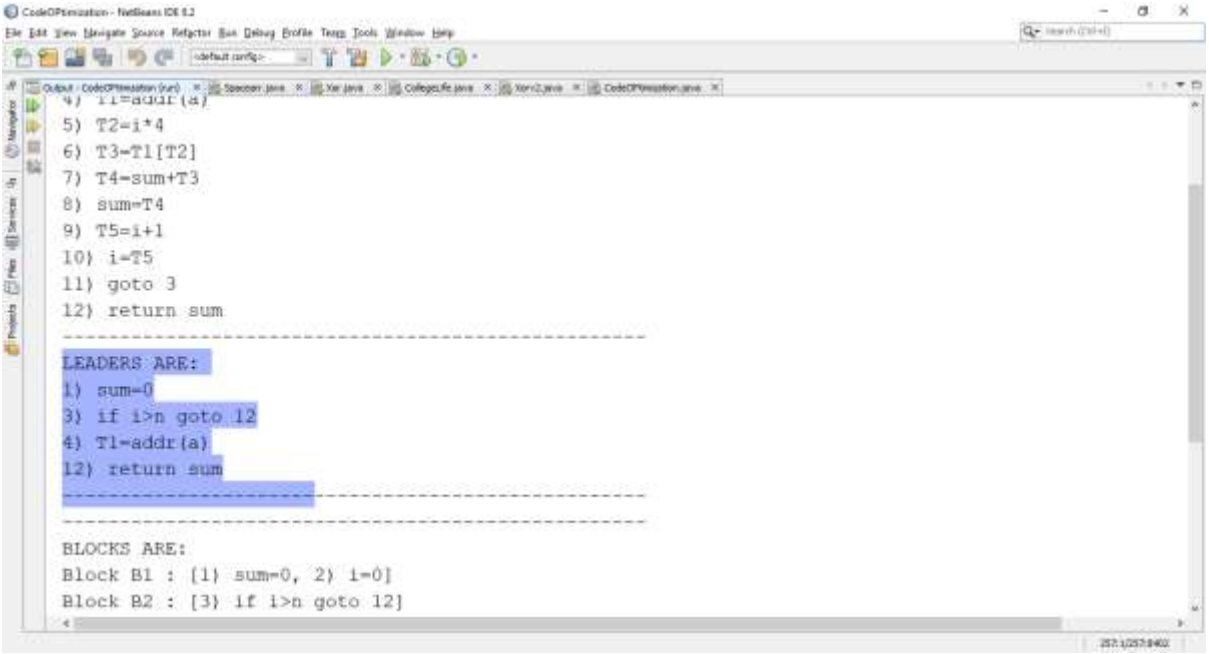
```
                for(int z=0; z<blocks.size();z++)
                {
                   list=blocks.get(z+1);
                    for(int m=0;m<list.size();m++)
                    {
                       if((list.get(m)+"").contains(in))
                       {
                           if((z+1)!=(i+1))
                          {
                            System.out.print(" B"+(z+1));
                          }
                       }
                    }
                }
                continue;
              }
            }


            System.out.println();
          }
            System.out.println("---------------------------------------------------- ");
       }
}
```

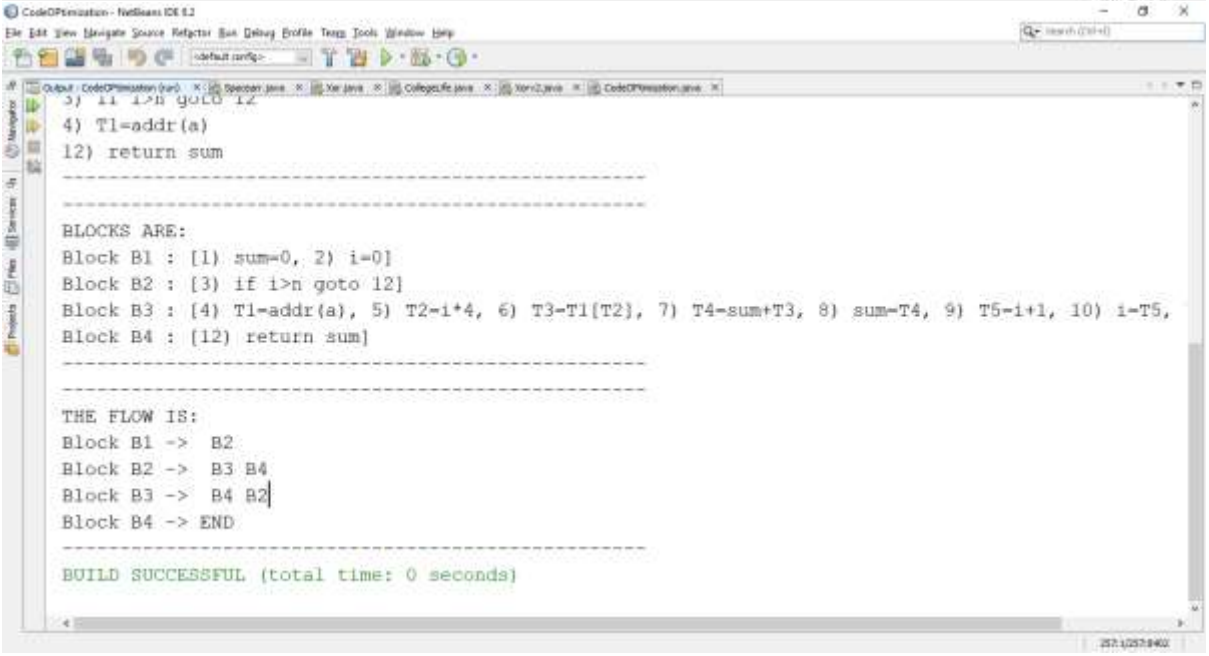
Input.txt


input.txt - Notepad
File Edit Format View Help
1) sum=0
2) i=0
3) if i>n goto 12
4) T1=addr(a)
5) T2=i*4
6) T3=T1[T2]
7) T4=sum+T3
8) sum=T4
9) T5=i+1
10) i=T5
11) goto 3
12) return sum
L 100% Windows (CRLF) UTF-8

OUTPUT:


run:
The file contains the following data:
1) sum=0
2) i=0
3) if i>n goto 12
4) T1=addr(a)
5) T2=i*4
6) T3=T1[T2]
7) T4=sum+T3
8) sum=T4
9) T5=i+1
10) i=T5
11) goto 3
12) return sum
LEADERS ARE:
1) sum=0
3) if i>n goto 12
4) T1=addr(a)


4) T1=addr(a)
5) T2=i*4
6) T3=T1[T2]
7) T4=sum+T3
8) sum=T4
9) T5=i+1
10) i=T5
11) goto 3
12) return sum
LEADERS ARE:
1) sum=0
3) if i>n goto 12
4) T1=addr(a)
12) return sum
BLOCKS ARE:
Block B1 : [1) sum=0, 2) i=0]
Block B2 : [3) if i>n goto 12]


4) T1=addr(a)
12) return sum
BLOCKS ARE:
Block B1 : [1) sum=0, 2) i=0]
Block B2 : [3) if i>n goto 12]
Block B3 : [4) T1=addr(a), 5) T2=i*4, 6) T3=T1[T2], 7) T4=sum+T3, 8) sum=T4, 9) T5=i+1, 10) i=T5,
Block B4 : [12) return sum]
THE FLOW IS:
Block B1 -> B2
Block B2 -> B3 B4
Block B3 -> B4 B2
Block B4 -> END
BUILD SUCCESSFUL (total time: 0 seconds)