**Practical No. 09**

**Name: Amarsingh Kashyap**

**Roll No: 101**

**Branch: CSE(2nd Shift)**

**AIM:** Write a program to perform loop detection by finding leader, basic blocks and program flow graph &amp; natural loop.

**CODE :**

dict={1 : **'count = 0'**,  
2 : **'Result = 0'**,  
3 : **'If count > 20 GOTO 8'**,  
4 : **'count = count + 1'**,  
5 : **'increment = 2 \* count'**,  
6 : **'result = result + increment'**,  
7 : **'GOTO 3'**,  
8 : **'end'**}  
  
key=[]  
value = []  
**for** keys, val **in** dict.items():  
 key.append(keys)  
 value.append(val)  
  
print(key,**"\n"**,value)  
  
leader = []  
block = {}  
pfg = []  
dom={}  
head=[]  
tail = []  
  
  
**def** findLeader():  
 idx = 0  
 **for** val **in** value:  
 **if** (idx == 0):  
 leader.append(val)  
 idx += 1  
 **continue** splitLst = val.split(**" "**)  
  
 **if** splitLst[0] == **'If'**:  
 leader.append(value[idx + 1])  
  
 length = len(splitLst)  
 **if** splitLst[length - 1].isdigit:  
 **if** splitLst[length - 2] == **'GOTO'**:  
 c = int(splitLst[length - 1])  
 leader.append(value[c - 1])  
  
 idx += 1  
  
  
**def** findPFG():  
 idx = 0  
  
 no = 1  
 c = 0  
 ifidx = 0  
  
 backedge = 0  
 **for** val **in** value:  
 *# print(val,"\n")* splitLst = val.split(**" "**)  
 length = len(splitLst)  
  
 **if** splitLst[0] == **'If'**:  
 lst = []  
 **for** j **in** range(0, idx):  
 lst.append(j + 1)  
  
 block[no] = lst  
 pfg.append(no)  
 pfg.append(no + 1)  
  
 *# pfg.append(no+1)* no += 1  
 block[no] = (idx + 1)  
 pfg.append(no)  
 pfg.append(no + 1)  
  
 backedge = idx  
 ifidx = idx + 1  
 no += 1  
 c = int(splitLst[length - 1])  
  
 idx += 1  
 **continue** idx += 1  
  
 lst = []  
 **for** j **in** range(ifidx + 1, idx):  
 lst.append(j)  
  
 block[no] = lst  
 pfg.append(no)  
 pfg.append(backedge)  
  
 no += 1  
 block[no] = c  
 pfg.append(backedge)  
 pfg.append(no)  
  
 **for** key **in** range(0, len(pfg), +2):  
 tail.append(pfg[key])  
 head.append(pfg[key + 1])  
  
  
**def** path():  
 **for** i **in** range(0, len(head)):  
 path = set()  
 lst = []  
 **if** head[i] **not in** dom:  
 **for** j **in** tail:  
 **if** tail[i] == j:  
 path.add(head[i])  
 path.add(j)  
 **break  
 else**:  
 path.add(j)  
 lst = list(path)  
 dom[head[i]] = lst  
  
 **pass** dom[1] = [1]  
  
  
*#Driver code*findLeader()  
findPFG()  
path()  
print(**"--------------------------------------------------------------------------------------------------"**)  
print(**"Leader statements are : "**)  
**for** i **in** leader:  
 print(i)  
print(**"--------------------------------------------------------------------------------------------------"**)  
  
print(**"Basic blocks are : \n"**)  
**for** key,val **in** block.items():  
 print(key,**" contains "**,val)  
  
print(**"--------------------------------------------------------------------------------------------------"**)  
  
print(**"Program flow graph is :"**)  
**for** key **in** range(0, len(pfg),+2):  
 print(pfg[key],**"-->"**,pfg[key+1])  
 tail.append(pfg[key])  
 head.append(pfg[key+1])  
print(**"--------------------------------------------------------------------------------------------------"**)  
  
print(**"Dominators are :"**)  
**for** key, val **in** dom.items():  
 print(key,**" : "**,val)  
print(**"--------------------------------------------------------------------------------------------------"**)  
i=0  
flag = 0  
print(**"Edge \t|Head \t|Tail \t Dom(Head) \t Dom(Tail)"**)  
**for** key **in** range(0, len(pfg),+2):  
 **if** head[i] **in** dom[tail[i]]:  
 print(pfg[key],**"-->"**,pfg[key+1],**"\t| "**,head[i],**"\t| "**,tail[i],**"\t"**,dom[head[i]],**"\t"**,dom[tail[i]],**" \t\tBackward Edge"**)  
 flag = 1  
 **else**:  
 print(pfg[key],**"-->"**,pfg[key+1],**"\t| "**,head[i],**"\t| "**,tail[i],**"\t"**,dom[head[i]],**"\t"**,dom[tail[i]],**" \t\tForward Edge"**)  
 i+=1  
print(**"--------------------------------------------------------------------------------------------------"**)  
**if** flag ==1:  
 print(**"Program contain Loop "**)  
**else**:  
 print(**"Program does not contain any Loop "**)  
  
  
i=0  
flag = 0  
print(**"Edge \t|Head \t|Tail \t Dom(Head) \t Dom(Tail)"**)  
**for** key **in** range(0, len(pfg),+2):  
 **if** head[i] **in** dom[tail[i]]:  
 print(pfg[key],**"-->"**,pfg[key+1],**"\t| "**,head[i],**"\t| "**,tail[i],**"\t"**,dom[head[i]],**"\t"**,dom[tail[i]],**" \tBackward Edge"**)  
 flag = 1  
 **else**:  
 print(pfg[key],**"-->"**,pfg[key+1],**"\t| "**,head[i],**"\t| "**,tail[i],**"\t"**,dom[head[i]],**"\t"**,dom[tail[i]],**" \tForward Edge"**)  
 i+=1  
**if** flag ==1:  
 print(**"Program contain Loop "**)  
**else**:  
 print(**"Program does not contain any Loop "**)

**OUTPUT :**





