**Practical No. 10**

**Name: Amarsingh Kashyap**

**Roll No: 101**

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

**Aim:** Write a program to generate the code using simple code generation algorithm.

**CODE:**

TAC={  
 **'t'**:[**'a'**, **'+'**, **'b'**],  
 **'u'**:[**'c'**,**'+'**,**'d'**],  
 **'v'**:[**'t'**,**'-'**,**'u'**],  
 **'w'**:[**'v'**,**'+'**,**'u'**]  
}  
lhs = {}  
rhs = []  
i=0  
stat = []  
**for** key,val **in** TAC.items():  
 stat.append(key)  
 lhs[key]=i  
 i+=1  
  
 rhs.append(val)  
  
print(lhs)  
print(rhs)  
  
instruction = []  
cost = 0  
R0=0  
R1=0  
  
**def** checkOperator(op):  
 **if** op==**'+'**:  
 **return 'ADD'  
 if** op==**'-'**:  
 **return 'SUB'  
 if** op==**'\*'**:  
 **return 'MUL'  
 if** op==**'/'**:  
 **return 'div'  
  
  
def** performOperation(operand1, operator, operand2):  
 operation = []  
 operation.append(checkOperator(operator))  
 operation.append(operand1)  
 operation.append(operand2)  
 instruction.append(operation)  
  
  
**def** addCost(type):  
 **if** type == **'RR'**:  
 **return** 1  
  
 **else**:  
 **return** 2  
  
  
i = 1  
rem = {**'R0'**: 1, **'R1'**: 1}  
index = -1  
**for** lst **in** rhs:  
 index += 1  
 particular = []  
 **if** lst[0] **not in** lhs **and** lst[2] **not in** lhs:  
 **for** key, val **in** rem.items():  
 **if** val == 1:  
 use = key  
 rem[key] = 0  
 **break** operand1 = lst[2]  
 operand2 = use  
 operator = lst[1]  
 particular.append(**'MOV'**)  
 particular.append(lst[0])  
 particular.append(operand2)  
 lhs[stat[index]] = use  
 cost = cost + addCost(**'RX'**)  
 instruction.append(particular)  
 performOperation(operand1, operator, operand2)  
 cost = cost + addCost(**'RX'**)  
  
  
  
 **elif** lst[0] **in** lhs **and** lst[2] **in** lhs:  
  
 operand1 = lhs[lst[0]]  
 rem[lhs[lst[0]]] = 1  
 operand2 = lhs[lst[2]]  
 operator = lst[1]  
 lhs[stat[index]] = operand1  
 performOperation(operand1, operator, operand2)  
 cost = cost + addCost(**'RR'**)  
  
print(rem)  
  
operand1 = key  
operand2 = stat[index]  
operator = **'Mov'**particular.append(operator)  
particular.append(operand1)  
particular.append(operand2)  
instruction.append(particular)  
*# performOperation(operand1, operator, operand2)*cost = cost + addCost(**'Rx'**)  
print(**"-----------------------------------------------------------------------------------------"**)  
print(**"Simple code instruction generated is : "**)  
**for** i **in** instruction:  
 print(i)  
print(**"-----------------------------------------------------------------------------------------"**)  
print(**"Total Cost of registers is : "**, cost)  
print(**"-----------------------------------------------------------------------------------------"**)

**OUTPUT:**

