Shri Ramdeobaba College of Engineering and Management, Nagpur - 440013

Department of Computer Science Engineering

Compiler Design Lab

PRACTICAL No. 7

Name : Shantanu Mane

Roll no. : 63

Topic : Code Generation
Platform : Windows or Linux

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

In [ ]:
```
TAC = {1 : "T1=a+b",
       2 : "T2=c+d",
       3 : "T3=T2-e",
       4 : "x=T1-T3"}
```

In [ ]:
```
OPCODE = {"+": "ADD",
          "-" : "SUB",
          "*" : "MUL",
          "/" : "DIV"}
```

In [ ]:
```
TAC
```

Out[ ]:
```
{1: 'T1=a+b', 2: 'T2=c+d', 3: 'T3=T2-e', 4: 'x=T1-T3'}
```

In [ ]:
```
OPCODE
```

Out[ ]:
```
{'+': 'ADD', '-': 'SUB', '*': 'MUL', '/': 'DIV'}
```

In [ ]:
```
def anyRegEmpty():
    for k,v in REGISTER_AVAILABLE.items():
        if v == 0:
            return True
    return False
```

In [ ]:
```
code = list()
REGISTER_AVAILABLE = {"R0" : 0, "R1": 0}
def generateCode():
    firstMoveDone = False
    index = 0
    for key,value in TAC.items():
        if not firstMoveDone:
                if anyRegEmpty():
                    emptyReg = ""
                    for i,j in REGISTER_AVAILABLE.items():
                        if j == 0:
```

```
                            emptyReg = i
                            REGISTER_AVAILABLE[i]=1
#                             print(REGISTER_AVAILABLE)
                            break
                operand = value.split("=")[1][0]
                nextOperand = value.split("=")[1][2]
                operation = value.split("=")[1][1]
                for l,k in OPCODE.items():
                    if l == operation:
                        opcode = k
                code.append(f"MOV {operand}, {emptyReg}")
                code.append(f"{opcode} {nextOperand}, {emptyReg}")
                firstMoveDone = True
        elif firstMoveDone:
            if anyRegEmpty:
                for i,j in REGISTER_AVAILABLE.items():
                    if j == 0:
                        emptyReg = i
                        index+=1
                        REGISTER_AVAILABLE[i] = 1
                        break
                print(value)
                operand = value.split("=")[1][0]
                nextOperand = value.split("=")[1][2]
                operation = value.split("=")[1][1]
                code.append(f"MOV {operand}, {emptyReg}")
                code.append(f"{opcode} {nextOperand}, {emptyReg}")
            print(value)
generateCode()
```

```
T2=c+d
T2=c+d
T3=T2-e
T3=T2-e
x=T1-T3
x=T1-T3
```

In [ ]:
```
code
```

Out[ ]:
```
['MOV a, R0',
 'ADD b, R0',
 'MOV c, R1',
 'ADD d, R1',
 'MOV T, R1',
 'ADD -, R1',
 'MOV T, R1',
 'ADD -, R1']
```

In [ ]:
```
REGISTER_AVAILABLE = {"R0" : 0, "R1": 0}
```

In [ ]:
```

```