In [3]:

```
input={
    1:'T1=a+b',
    2:'T2=c+d',
    3:'T3=T2-e',
    4:'x=T1-T3'
}
compiler_vars=['T1','T2','T3']
user_vars=['a','b','c','d','e','x']
reg_desc={
    'R0':'0',
    # 'R1':'0',
    # 'R2':'0',
    # 'R3':'0',
}
addr_desc={
    '00':'0',
    '01':'0',
    '10':'0',
    '11':'0',
}
opcode={
    '+':'ADD',
    '-':'SUB',
    '*':'MUL',
    '/':'DIV'
}
```

In [4]:

```
def getreg(statement):
    des=''
    oper1=''
    oper2=''
    oper3=''
    op=''
    L=''
    for i in compiler_vars:
        index=statement.find(i)
        # print(i,index)
        if index==0:
            des=i
        if index in [2,3]:
            oper1=i
        if index in [4,5]:
            oper2=i
    for i in user_vars:
        index=statement.find(i)
        # print(i,index)
        if index==0:
            des=i
        if index in [2,3]:
            oper1=i
        if index in [4,5]:
            oper2=i
    for i in list(opcode.keys()):
        if i in statement:
            op=i
    flag=False
    for i in list(reg_desc.keys()):
        if reg_desc[i]=='0':
            reg_desc[i]=oper1
            L=i
            flag=True
            break
  # print(flag)
    if flag==False:
        for i in list(addr_desc.keys()):
            if addr_desc[i]=='0':
                addr_desc[i]=oper1
                L=i
                break
    return des,oper1,op,oper2,L
```

In [5]:

```
def free_reg(reg_desc,curr_statement):
    flag=False
    last_statement=list(input.keys())[-1]
    future_statements=[]
    for i in range(curr_statement,last_statement+1):
        future_statements.append(input[i])
    for i in list(reg_desc.keys()):
        for j in future_statements:
            if reg_desc[i] in j:
                flag=True
    if(not flag):
        reg_desc[i]='0'
```

In [9]:

```
free_reg(reg_desc,2)
```

In [10]:

```
def gencode(input):
    machine_code=[]
    curr_statement=1
    for i in list(input.keys()):
        free_reg(reg_desc,curr_statement)
        des,oper1,op,oper2,L=getreg(input[i])
        print( des,oper1,op,oper2,L)
        if(oper1!=L):
            machine_code.append('MOV '+oper1+' '+L)
        machine_code.append(opcode[op]+' '+oper2+' '+L)
        curr_statement+=1
    return machine_code
```

In [11]:

```
machine_code = gencode(input)
```

```
T1 a + b 00
T2 c + d 01
T3 T2 -  10
x T1 - T3 11
```

In [12]:

```
def print_code(machine_code):
    for i in machine_code:
        print(i)
```

In [13]:

```
print_code(machine_code)
```

```
MOV a 00
ADD b 00
MOV c 01
ADD d 01
MOV T2 10
SUB  10
MOV T1 11
SUB T3 11
```

In [ ]: