

## Laboratory-6

### Question

Perform local optimization on a basic block.

### Python File:

```
import pandas as pd
a=pd.read_csv("input.csv")
c=a.shape
print(a)
b=[]
for i in range(c[0]):
    for j in range(i+1,c[0]):
        if(a['right'][i]==a['right'][j]):
            for d in range(c[0]):
                b=b+[len(a['right'][d])]
                for z in range(b[d]):
                    if(a['right'][d][z]==a['left'][j]):
                        x=list(a['right'][d])
                        x[z]=a['left'][i]
                        l=''.join(x)

a['right'][d]=a['right'][d].replace(a['left'][j],a['left'][i])
a['left'][j]=a['left'][i]
df=pd.DataFrame(a)
df.to_csv('output1.csv',index=False)
p=pd.read_csv("output1.csv")
print("After checking and putting the value of common expression ")
print(p)
i=0
j=i+1
while(j<c[0]):
    if(p['right'][i]==p['right'][j]):
        if(p['left'][i]==p['left'][j]):
            p.drop([j],axis=0,inplace=True)
            i+=2
            j+=1
        else:
            i+=1
    j+=1
    if(j==c[0]):
        i=i+1
        j=i+1
    if(i==c[0]):
        j=c[0]
print("After elemenating the common expression")
df=pd.DataFrame(p)
df.to_csv('output1.csv',index=False)
p=pd.read_csv("output1.csv")
print(p)
c=p.shape
#print(c)
count=0
i=0
```

```

j=0
h=1
while(j<c[0] and i<c[0]):
    b=[]
    b=b+[(len(p['right'][j]))]
    for z in range(b[0]):
        if(p['right'][j][z]==p['left'][i]):
            count=1
    j+=1
    if(j==c[0]):
        if(count!=1):
            p.drop([i],axis=0,inplace=True)
            df=pd.DataFrame(p)
            df.to_csv('output1.csv',index=False)
            p=pd.read_csv("output1.csv")
            c=p.shape
            print(c)
        i+=1
        j=0
print("After dead code elimination")
print(p)
df=pd.DataFrame(p)
df.to_csv('output1.csv',index=False)
p=pd.read_csv("output1.csv")
c=p.shape
print("The final optimized code is....")
for i in range(c[0]):
    print(str(p['left'][i])+"="+str(p['right'][i]))

```

**Output:**

```

● kal-el@mos-13:~/Desktop/Compilers/Compilers Lab/lab_5$ ./generator < input.txt
x = 4
count = 1
if x < 5 goto L2
goto L3
L2:
t0 = count * 2
count = t0
goto G0
L3:
t1 = count + 2
count = t1
G0:
completed!

```

**CSV File:**

```

left,right
a,9
b,c+d
e,c+d
f,b+e
r,f

```

**Result:**

Intermediate code for conditional and looping constructs was generated successfully.