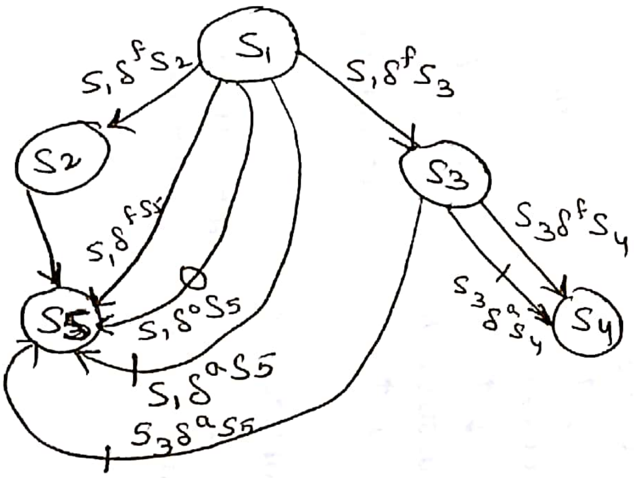
~~HOW~~
HOMEWORK-4

SHIVANI SABHLOK
500237896

(1)

	Read	Write	Dependency
S1 :	a	a	
S2 :	a, b	b	$S_1 \delta^f S_2$
S3 :	a, d	c	$S_1 \delta^f S_3$
S4 :	c	d	$S_3 \delta^f S_4$, $S_3 \delta^a S_4$
S5 :	a	a	$S_1 \delta^f S_5$, $S_1 \delta^o S_5$, $S_1 \delta^a S_5$, $S_2 \delta^a S_5$, $S_3 \delta^a S_5$



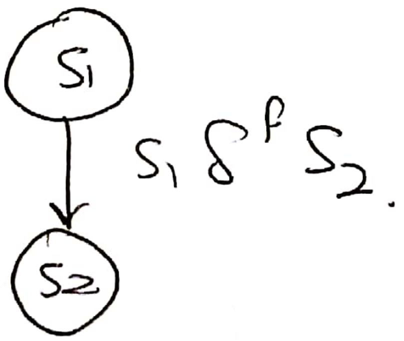
(2) for (i = 0; i < max, i++)

for (j = 0; j < Max; j++)

S1 : A[i][j] = A[i+1][j-1]

S2 : B[i][j] = A[i][j]

Sol^n



	Dist Vector	Direction Vector			
S_1	$(-1, 1)$	$(-, +)$	$\Rightarrow$	$-$	$\Rightarrow$ Anti-dependency.
S_2	NA	NA			

- if we reorder S1, sign of the direction vector would change $\Rightarrow$ no reordering possible.
- for S2, since there are no dependencies, 2! re-orderings are allowed. There will be no memory conflicts.

Thus, loops cannot be reordered because it will change the sign of direction vector for S1.

3.

```
for (i=0, i< Max; i++)
    for (j=0; j< max; j++)
        for (k=0; k< Max; k++)
            for (M=0; M< Max ; M++)
                BODY (i,j, k, M)
```

Soln

(a) A(i)(j)(k)(M) = B(i-1)(j-1)(k-1)(M-1)

since the two are independent memory locations, all combinations of loop ordering are allowed

$\Rightarrow$ 4! reordering combinations are allowed

(b) $A(i)(j)(k)(m) = A(i-1)(j)(k)(m)$

Direction Vector : $(+, 0, 0, 0)$

Dist. vector : $(1, 0, 0, 0)$

Since direction vector will always be +ve, all combinations of reordering are allowed

$\Rightarrow$ 4! (= 24) possible reorderings are possible

(c) $A(i)(j)(k)(M+1) = A(i)(j)(k)(M) + B(i)$

Direction vector for A : $(0, 0, 0, +)$

Distance vector for A : $(0, 0, 0, 1)$

B has no effect here

Thus, since the sign of the direction vector won't change, 4! (=24) reorderings are possible.