**ALGORITHM DESIGN**

**Safety Algorithm:**

1) Let Work and Finish be Arrays of length ‘m’ and ‘n’ respectively.

Initialize: Work = Available

Finish[i] = false; for i=1, 2, 3, 4….n

2) Find an i such that both

a) Finish[i] = false

b) Needi <= Work

if no such i exists goto step (4)

3) Work = Work + Allocation[i]

Finish[i] = true

goto step (2)

4) if Finish [i] = true for all i

then the system is in a safe state

**Request Algorithm:**

1) If Request i <= Need i

Goto step (2) ; otherwise, raise an error condition, since the process has exceeded its maximum claim.

2) If Request i <= Available

Goto step (3); otherwise, Pi must wait, since the resources are not available.

3) Have the system pretend to have allocated the requested resources to process Pi by modifying the state as

follows:

Available = Available – Request i

Allocation i = Allocation i +Request i

 Need i = Need i– Request i