ALGo –

1. Specify the header files and various variables.

2. Enter number of resources and processes.

3. Enter all other required input such as Allocated resource table and maximum claim table.

4. Enter available resources as

 for (i = 0; i < resources; i++)

    {

            available[i] = max[i] - allocation[i];

    }

        printf("\nAvailable resources:");

        for (i = 0; i < resources; i++)

    {

            printf("\t%d", available[i]);

        }

5. if maximum[i][j] - current[i][j] > available[j] execute the process

                                running[i] = 0;

                                counter--;

                                safe = 1;

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

                    {

                                    available[j] += current[i][j];

                                }

6. if !safe the process is in unsafe state otherwisw safe.

7. Terminate the program by returning 0.

**INPUT -**

Enter the number of processes3

Enter the number of resources:2

Enter Claim Vector:3

4

Enter Allocated Resource Table:

1

2

3

4

5

6

Enter Maximum Claim Table:

1

23

4

56

3

23

The Claim Vector is:    3       4

The Allocated Resource Table is:

        1       2

        3       4

        5       6

The Maximum Claim Table is:

        41      23

        4       56

        3       23

Allocated resources:    9       12

Available resources are:        -6      -8

**OUTPUT -**

The processes are in unsafe state.

#include <stdio.h>

int current[10][10], maximum[10][10], available[10];

int allocation[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

int max[10], running[10], safe = 0;

int counter = 0, i, j, exec, resources, processes, k = 1;

int main()

{

    printf("\nEnter number of processes: ");

        scanf("%d", &processes);

        for (i = 0; i < processes; i++)

    {

            running[i] = 1;

            counter++;

        }

        printf("\nEnter number of resources: ");

        scanf("%d", &resources);

        printf("\nEnter Claim Vector:");

        for (i = 0; i < resources; i++)

    {

            scanf("%d", &max[i]);

        }

       printf("\nEnter Allocated Resource Table:\n");

        for (i = 0; i < processes; i++)

    {

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

        {

              scanf("%d", &current[i][j]);

            }

        }

        printf("\nEnter Maximum Claim Table:\n");

        for (i = 0; i < processes; i++)

    {

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

        {

                    scanf("%d", &maximum[i][j]);

            }

        }

    printf("\nThe Claim Vector is: ");

        for (i = 0; i < resources; i++)

    {

            printf("\t%d", max[i]);

    }

        printf("\nThe Allocated Resource Table:\n");

        for (i = 0; i < processes; i++)

    {

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

        {

                    printf("\t%d", current[i][j]);

            }

        printf("\n");

        }

        printf("\nThe Maximum Claim Table:\n");

        for (i = 0; i < processes; i++)

    {

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

        {

                printf("\t%d", maximum[i][j]);

            }

            printf("\n");

        }

        for (i = 0; i < processes; i++)

    {

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

        {

                    allocation[j] += current[i][j];

            }

        }

        printf("\nAllocated resources:");

        for (i = 0; i < resources; i++)

    {

            printf("\t%d", allocation[i]);

        }

        for (i = 0; i < resources; i++)

    {

            available[i] = max[i] - allocation[i];

    }

        printf("\nAvailable resources:");

        for (i = 0; i < resources; i++)

    {

            printf("\t%d", available[i]);

        }

        printf("\n");

        while (counter != 0)

    {

            safe = 0;

            for (i = 0; i < processes; i++)

        {

                    if (running[i])

            {

                        exec = 1;

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

                {

                                if (maximum[i][j] - current[i][j] > available[j])

                    {

                                    exec = 0;

                                    break;

                                }

                        }

                        if (exec)

                {

                                printf("\nProcess%d is executing\n", i + 1);

                                running[i] = 0;

                                counter--;

                                safe = 1;

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

                    {

                                    available[j] += current[i][j];

                                }

                            break;

                        }

                    }

            }

            if (!safe)

        {

                    printf("\nThe processes are in unsafe state.\n");

                    break;

            }

        else

        {

                    printf("\nThe process is in safe state");

                    printf("\nAvailable vector:");

                    for (i = 0; i < resources; i++)

            {

                        printf("\t%d", available[i]);

                    }

                printf("\n");

            }

        }

        return 0;

}