EXPERIMENT 13
DEADLOCK PREVENTION

AIM
Write a C program to simulate algorithm for Deadlock Prevention

ALGORITHM FOR INPUT()
STEP 0: START
STEP 1: Prompt the user to enter the number of process (n) and number of resource instances (r)
STEP 2: ~~END~~ Prompt user to enter the Max Matrix values for each process and resource combination and store in max array
STEP 3: Prompt user to enter the Allocation Matrix values for each process and resource combination and store it in 'alloc' array.
STEP 4: Prompt the user to enter the available resources and store it in the 'avail' array.

ALGORITHM FOR SHOW()
STEP 0: START
STEP 1: Print the header with columns for Process, Allocation, Max and Available resources.

STEP 2: To display process info.
from i=1 till n
i) Print process number P[i].
ii) Print Allocation values
iii) Print Max values
iv) Display Available resource

ALGORITHM FOR CAL()
STEP 0: START
STEP 1: For each process i=0 till n-1, and each resource j=0 till r-1, calculate Need Matrix value as
Need[i][j] = Max[i][j] - Alloc[i][j]
STEP 2: Initialize 'finish' array to mark finished process
STEP 3: Initialize 'flag' to 1 to indicate deadlock detection loop.
while flag is true,
i) Set flag = 0
ii) For each process i=0 till n-1
Initialize counter c to 0.
iii) For each process j=0 till r-1
- If Need[i][j] <= avail[j].
increment c.
- If c equals r, release resources, mark process as finished, update available resources and set flag to 1.

STEP 4: Check if any process is still

unfinished.
i) If present then, set 'deadlock' flag to 1 and display deadlock process
ii) If no deadlock is detected, display "NO Deadlock Occurs".
STEP 5: END

ALGORITHM FOR MAIN()
STEP 0: START
STEP 1: Call the necessary functions for deadlock prevention, INPUT(), SHOW(), CALL().
STEP 2: END.

RESULT
Experiment executed successfully & output obtained.