10. Consider the following process table with number of processes that contains allocation field (for showing the number of resources of type: A, B and C allocated to each process in the table), max field (for showing the maximum number of resources of type: A, B, and C that can be allocated to each process). Write a program to calculate the entries of need matrix using the formula: (Need)i = (Max)i - (Allocation)i in c Process Allocation Max Availble A B C A B C A B C P0 1 1 2 5 4 4 3 2 1 P1 2 1 2 4 3 3 P2 3 0 1 9 1 3 P3 0 2 0 8 6 4 P4 1 1 2 2 2 3

#include<stdio.h>

#define NUM\_PROCESSES 5

#define NUM\_RESOURCES 3

int main()

{

int allocation[NUM\_PROCESSES][NUM\_RESOURCES] = {

{1, 1, 2},

{2, 1, 2},

{3, 0, 1},

{0, 2, 0},

{1, 1, 2}

};

int max[NUM\_PROCESSES][NUM\_RESOURCES] = {

{5, 4, 4},

{4, 3, 3},

{9, 1, 3},

{8, 6, 4},

{2, 2, 3}

};

int available[NUM\_RESOURCES] = {3, 2, 1};

int need[NUM\_PROCESSES][NUM\_RESOURCES];

// Calculate the need matrix

for(int i = 0; i < NUM\_PROCESSES; i++) {

for(int j = 0; j < NUM\_RESOURCES; j++) {

need[i][j] = max[i][j] - allocation[i][j];

}

}

// Print the need matrix

printf("Need matrix:\n");

for(int i = 0; i < NUM\_PROCESSES; i++) {

printf("P%d: ", i);

for(int j = 0; j < NUM\_RESOURCES; j++) {

printf("%d ", need[i][j]);

}

printf("\n");

}

return 0;

}

