#include <stdio.h>

1. **int** main()
2. {
3. // P0, P1, P2, P3, P4 are the Process names here
5. **int** n, m, i, j, k;
6. n = 5;                         // Number of processes
7. m = 3;                         // Number of resources
8. **int** alloc[5][3] = {{0, 1, 0},  // P0 // Allocation Matrix
9. {2, 0, 0},  // P1
10. {3, 0, 2},  // P2
11. {2, 1, 1},  // P3
12. {0, 0, 2}}; // P4
14. **int** max[5][3] = {{7, 5, 3},  // P0 // MAX Matrix
15. {3, 2, 2},  // P1
16. {9, 0, 2},  // P2
17. {2, 2, 2},  // P3
18. {4, 3, 3}}; // P4
20. **int** avail[3] = {3, 3, 2}; // Available Resources
22. **int** f[n], ans[n], ind = 0;
23. **for** (k = 0; k < n; k++)
24. {
25. f[k] = 0;
26. }
27. **int** need[n][m];
28. **for** (i = 0; i < n; i++)
29. {
30. **for** (j = 0; j < m; j++)
31. need[i][j] = max[i][j] - alloc[i][j];
32. }
33. **int** y = 0;
34. **for** (k = 0; k < 5; k++)
35. {
36. **for** (i = 0; i < n; i++)
37. {
38. **if** (f[i] == 0)
39. {
40. **int** flag = 0;
41. **for** (j = 0; j < m; j++)
42. {
43. **if** (need[i][j] > avail[j])
44. {
45. flag = 1;
46. **break**;
47. }
48. }
49. **if** (flag == 0)
50. {
51. ans[ind++] = i;
52. **for** (y = 0; y < m; y++)
53. avail[y] += alloc[i][y];
54. f[i] = 1;
55. }
56. }
57. }
58. }
59. **int** flag = 1;
60. **for** (**int** i = 0; i < n; i++)
61. {
62. **if** (f[i] == 0)
63. {
64. flag = 0;
65. printf("The following system is not safe");
66. **break**;
67. }
68. }
69. **if** (flag == 1)
70. {
71. printf("Following is the SAFE Sequence\n");
72. **for** (i = 0; i < n - 1; i++)
73. printf(" P%d ->", ans[i]);
74. printf(" P%d", ans[n - 1]);
75. }
76. **return** (0);
77. }

