Code:

```
#include <cstring>
#include <iostream>

using namespace std;

bool findSafeSequence(int max[][10], int allocated[][10], int available[],
                      int m, int n, int safeSeq[]) {

  int work[10];
  bool finish[10];

  for (int i = 0; i < m; i++)
    work[i] = available[i];  // 3 3 2

  for (int i = 0; i < n; i++)
    finish[i] = false;

  int count = 0;

  while (count < n) {  // 0 < 5
    bool found = false;
    for (int i = 0; i < n; i++)  // 1
      if (!finish[i]) {
        int j;
        for (j = 0; j < m; j++) {
          if (max[i][j] - allocated[i][j] > work[j]) // 1 2 2 > 4 5 4
            break;
        }
        if (j == m) {  //  3 = 3
          for (int k = 0; k < m; k++) {
            work[k] += allocated[i][k];  /// 3 3 2 + 1 2 2 = 4 5 4
          }
          safeSeq[count++] = i;
          finish[i] = true;
          found = true;
        }
      }
    if (!found)
      return false;
  }

  return true;
}

int main() {
  int n, m;
```

```
  cout << "Enter the number of processes: ";
  cin >> n;
  cout << "\nEnter the number of resources: ";
  cin >> m;

  int max[10][10];
  int allocated[10][10];
  int available[10];
  int safeSeq[10];

  cout << "\nEnter the maximum resource requirements for each process: "
       << endl;
  for (int i = 0; i < n; i++)
    for (int j = 0; j < m; j++)
      cin >> max[i][j];

  cout << "\nEnter the allocated resource for each process: " << endl;
  for (int i = 0; i < n; i++)
    for (int j = 0; j < m; j++)
      cin >> allocated[i][j];

  cout << "\nEnter the available resources: ";
  for (int i = 0; i < m; i++)
    cin >> available[i];

  if (findSafeSequence(max, allocated, available, m, n, safeSeq)) {
    cout << "\nSafe sequence: ";
    for (int i = 0; i < n; i++)
      cout << "P" << safeSeq[i] << " ";
    cout << endl;
  } else
    cout << "\nNo safe sequence exists." << endl;

  return 0;
}
```

Output:

Enter the number of processes: 5

Enter the number of resources: 4

Enter the maximum resource requirements for each process:
5 1 1 7
3 2 1 1
3 3 2 1
4 6 1 2
6 3 2 5

Enter the allocated resource for each process:
3 0 1 4
2 2 1 0
3 1 2 1
0 5 1 0
4 2 1 2

Enter the available resources: 0 3 0 1

No safe sequence exists.