```c
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>
#define P 4
#define R 4
int main() {
int available[R];
int allocated[P][R];
int max_required[P][R];
int need[P][R];
bool finish[P] = {false};
int sequence[P];
printf("Enter Resources allocated to each process:\n");
for (int i = 0; i < P; i++) {
printf("Resources of process %d: ", i + 1);
for (int j = 0; j < R; j++) {
scanf("%d", &allocated[i][j]);
}
}
printf("Enter maximum resources required for each process:\n");
for (int i = 0; i < P; i++) {
printf("Maximum resources required for process %d: ", i + 1);
for (int j = 0; j < R; j++) {
scanf("%d", &max_required[i][j]);
}
}
printf("Enter available resources: ");
for (int i = 0; i < R; i++) {
scanf("%d", &available[i]);
}
for (int i = 0; i < P; i++) {
for (int j = 0; j < R; j++) {
need[i][j] = max_required[i][j] - allocated[i][j];
}
}
printf("Need of resources required for each process:\n");
for (int i = 0; i < P; i++) {
printf("Need of resources required for process %d: ", i + 1);for (int j = 0; j < R; j++) {
printf("%d ", need[i][j]);
}
printf("\n");
}
int count = 0;
while (count < P) {
bool found = false;
for (int i = 0; i < P; i++) {
if (!finish[i]) {
bool canProceed = true;
for (int j = 0; j < R; j++) {
if (need[i][j] > available[j]) {
canProceed = false;
break;
```

```c
}
}
if (canProceed) {
for (int j = 0; j < R; j++) {
available[j] += allocated[i][j];
}
sequence[count++] = i;
finish[i] = true;
found = true;
}
}
}
if (!found) {
printf("System is not in a safe state.\n");
return 0;
}
}
printf("System is in a safe state.\n");
printf("Safe sequence is: ");
for (int i = 0; i < P; i++) {
printf("P%d ", sequence[i] + 1);
}
printf("\n");
return 0;
}
```