```c
#include <stdio.h>
#define MAX_PROCESSES 5
#define MAX_RESOURCES 3
int is_safe();
int available[MAX_RESOURCES] = {3, 3, 2};
int maximum[MAX_PROCESSES][MAX_RESOURCES] = {
    {7, 5, 3},
    {3, 2, 2},
    {9, 0, 2},
    {2, 2, 2},
    {4, 3, 3}
};
int allocation[MAX_PROCESSES][MAX_RESOURCES] = {
    {0, 1, 0},
    {2, 0, 0},
    {3, 0, 2},
    {2, 1, 1},
    {0, 0, 2}
};
int request_resources(int process_num, int request[]) {
    for (int i = 0; i < MAX_RESOURCES; i++) {
        if (request[i] > available[i] || request[i] > maximum[process_num][i] - allocation[process_num][i])
            return 0;
    }
    for (int i = 0; i < MAX_RESOURCES; i++) {
        available[i] -= request[i];
        allocation[process_num][i] += request[i];
```

```
    }
    if (is_safe()) {
        return 1;
    } else {
        for (int i = 0; i < MAX_RESOURCES; i++) {
            available[i] += request[i];
            allocation[process_num][i] -= request[i];
        }
        return 0;
    }
}
int is_safe() {
    int work[MAX_RESOURCES];
    int finish[MAX_PROCESSES] = {0};
    for (int i = 0; i < MAX_RESOURCES; i++)
        work[i] = available[i];
    int count = 0;
    while (count < MAX_PROCESSES) {
        int found = 0;
        for (int i = 0; i < MAX_PROCESSES; i++) {
            if (!finish[i]) {
                int j;
                for (j = 0; j < MAX_RESOURCES; j++) {
                    if (maximum[i][j] - allocation[i][j] > work[j])
                        break;
                }
```

```
            if (j == MAX_RESOURCES) {
                for (int k = 0; k < MAX_RESOURCES; k++)
                    work[k] += allocation[i][k];
                finish[i] = 1;
                found = 1;
                count++;
            }
        }
    }
    if (!found)
        return 0;
  }
  return 1;
}
int main() {
  int process_num, request[MAX_RESOURCES];
  printf("Enter process number (0-4): ");
  scanf("%d", &process_num);
  printf("Enter resource request (e.g., 0 1 0): ");
  for (int i = 0; i < MAX_RESOURCES; i++)
    scanf("%d", &request[i]);
  if (request_resources(process_num, request))
    printf("Request granted.\n");
  else
    printf("Request denied. System is not in safe state.\n");
  return 0;
}
```

```
Enter process number (0-4): 2
Enter resource request (e.g., 0 1 0): 1
0
0
Request granted.

--------------------------------
Process exited after 116.5 seconds with return value 0
Press any key to continue . . .
```