

## Code:

```
#include <stdio.h>
#include <stdlib.h>
#include <stdbool.h>

int n = 5;
struct resource{
    int a, b, c;
};

bool check(struct resource avail, struct resource
res){
    if(avail.a>=res.a && avail.b>=res.b &&
avail.c>=res.c){
        return true;
    }
    return false;
}

void bankers(int c, struct resource allocated[],
struct resource need[], struct resource avail){
    int completed[n];
    for(int i=0; i<n; i++) completed[i]=1;
    int counter = 0, completed_count=0;

    struct resource remanining[c];
    for(int i=0; i<c; i++){
        remanining[i].a = need[i].a - allocated[i].a;
        remanining[i].b = need[i].b - allocated[i].b;
        remanining[i].c = need[i].c - allocated[i].c;
    }
    printf("\n\nProcess execution seq: \n");
    while(completed_count!=c){

        if(completed[counter]!=0 && check(avail,
remanining[counter])){
            remanining[counter].a = 0;
            remanining[counter].b = 0;
            remanining[counter].c = 0;
            avail.a += allocated[counter].a;
            avail.b += allocated[counter].b;
            avail.c += allocated[counter].c;
            completed[counter] = 0;
            completed_count++;
            printf("Process %d\n", counter);
        }

        counter = (counter+1)%c;
    }
}
```

```
int main(){
    printf("Enter number of process: ");
    scanf("%d", &n);

    struct resource res;
    printf("Enter number of resources for a, b, c:
");
    scanf("%d %d %d", &res.a, &res.b, &res.c);
    struct resource avail = res;
    printf("\n");

    struct resource allocated[n];
    for(int i=0; i<n; i++){
        printf("Enter allocation for process %d (a,
b, c): ", (i+1));
        scanf("%d", &allocated[i].a);
        scanf("%d", &allocated[i].b);
        scanf("%d", &allocated[i].c);
        avail.a -= allocated[i].a;
        avail.b -= allocated[i].b;
        avail.c -= allocated[i].c;
    }

    struct resource need[n];
    for(int i=0; i<n; i++){
        printf("Enter need for process %d (a, b, c):
", (i+1));
        scanf("%d", &need[i].a);
        scanf("%d", &need[i].b);
        scanf("%d", &need[i].c);
    }

    bankers(n, allocated, need, avail);

    return 0;
}
```

## **Output:**

@somes4545 → /workspaces/TE-Labs/OSL

(main) \$ g++ 5.c && ./a.out

Enter number of process: 5

Enter number of resources for a, b, c: 10 6 7

Enter allocation for process 1 (a, b, c): 1 1 2

Enter allocation for process 2 (a, b, c): 2 1 2

Enter allocation for process 3 (a, b, c): 4 0 1

Enter allocation for process 4 (a, b, c): 0 2 0

Enter allocation for process 5 (a, b, c): 1 1 2

Enter need for process 1 (a, b, c): 4 3 3

Enter need for process 2 (a, b, c): 3 2 2

Enter need for process 3 (a, b, c): 9 0 2

Enter need for process 4 (a, b, c): 7 5 3

Enter need for process 5 (a, b, c): 1 1 2

Process execution seq:

Process 1

Process 4

Process 0

Process 2

Process 3