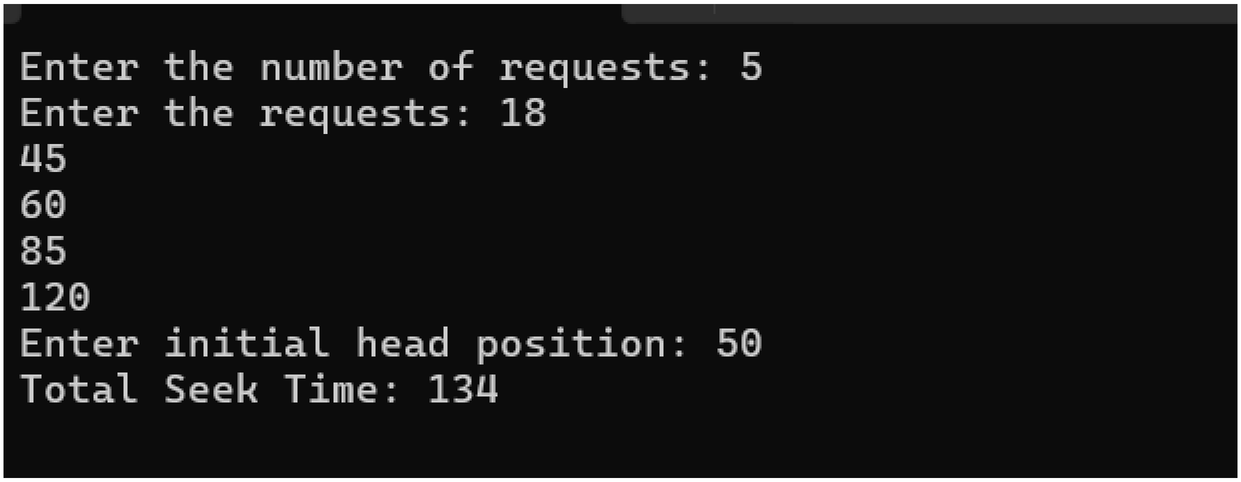
1

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n, head, total_seek = 0;
    printf("Enter the number of requests: ");
    scanf("%d", &n);
    int requests[n];
    printf("Enter the requests: ");
    for (int i = 0; i < n; i++) {
        scanf("%d", &requests[i]);
    }
    printf("Enter initial head position: ");
    scanf("%d", &head);
    for (int i = 0; i < n; i++) {
        total_seek += abs(requests[i] - head);
        head = requests[i];
    }

    printf("Total Seek Time: %d\n", total_seek);
    return 0;
}
```

Output:



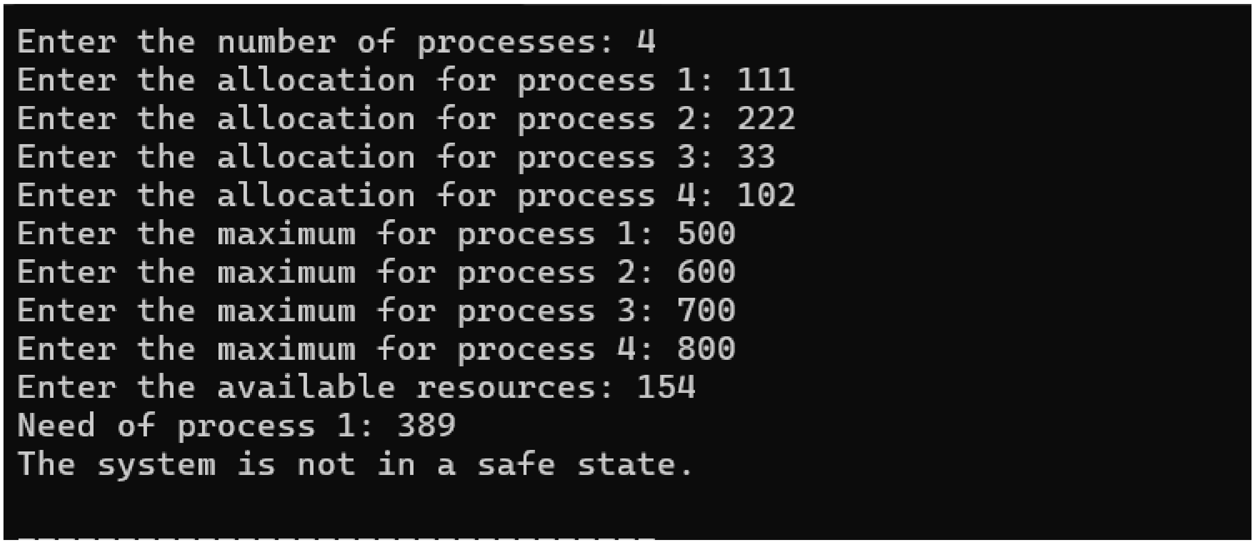
```
Enter the number of requests: 5
Enter the requests: 18
45
60
85
120
Enter initial head position: 50
Total Seek Time: 134
```

2 =

m

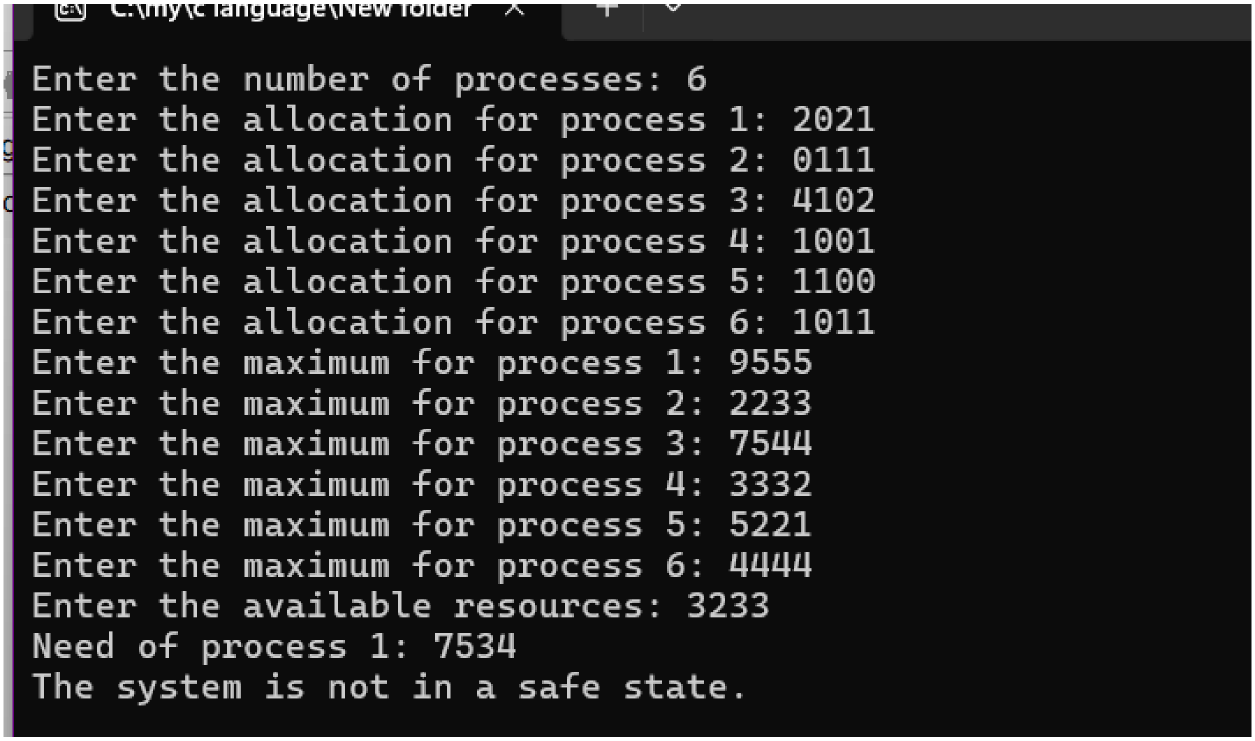
```
#include <stdio.h>
int main() {
    int n, i, work;
    printf("Enter the number of processes: ");
    scanf("%d", &n);
    int allocation[n], max[n], need[n];
    int available;
    for (i = 0; i < n; i++) {
        printf("Enter the allocation for process %d: ", i + 1);
        scanf("%d", &allocation[i]);
    }
    for (i = 0; i < n; i++) {
        printf("Enter the maximum for process %d: ", i + 1);
        scanf("%d", &max[i]);
    }
    printf("Enter the available resources: ");
    scanf("%d", &available);

    work = available;

    for (i = 0; i < n; i++) {
        need[i] = max[i] - allocation[i];
        printf("Need of process %d: %d\n", i + 1, need[i]);

        if (need[i] <= work) {
            work += allocation[i];
            printf("Allocated resources to process %d, updated work: %d\n", i + 1, work);
        } else {
            printf("The system is not in a safe state.\n");
            return 0;
        }
    }

    printf("System is in a safe state.\n");
    return 0;
}
```

Output:



```
Enter the number of processes: 4
Enter the allocation for process 1: 111
Enter the allocation for process 2: 222
Enter the allocation for process 3: 33
Enter the allocation for process 4: 102
Enter the maximum for process 1: 500
Enter the maximum for process 2: 600
Enter the maximum for process 3: 700
Enter the maximum for process 4: 800
Enter the available resources: 154
Need of process 1: 389
The system is not in a safe state.
```

```
3. #include <stdio.h>


void firstFit(int blockSize[], int m, int processSize[], int n) {
    int allocation[n];


    for (int i = 0; i < n; i++) {
        allocation[i] = -1;
    }


    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            if (blockSize[j] >= processSize[i]) {
                allocation[i] = j;
                blockSize[j] -= processSize[i];
                break;
            }
        }
    }
    printf("\nProcess No.\tProcess Size\tBlock No.\n");
    for (int i = 0; i < n; i++) {
        printf(" %d\t\t%d\t\t", i + 1, processSize[i]);
        if (allocation[i] != -1) {
            printf("%d\n", allocation[i] + 1);
        } else {
            printf("Not Allocated\n");
        }
    }
}

int main() {
    int blockSize[] = {100, 170, 40, 205, 300,185};
    int processSize[] = {200,15,185,75,175,80};
    int m = sizeof(blockSize) / sizeof(blockSize[0]);
    int n = sizeof(processSize) / sizeof(processSize[0]);

    firstFit(blockSize, m, processSize, n);

    return 0;
}
```

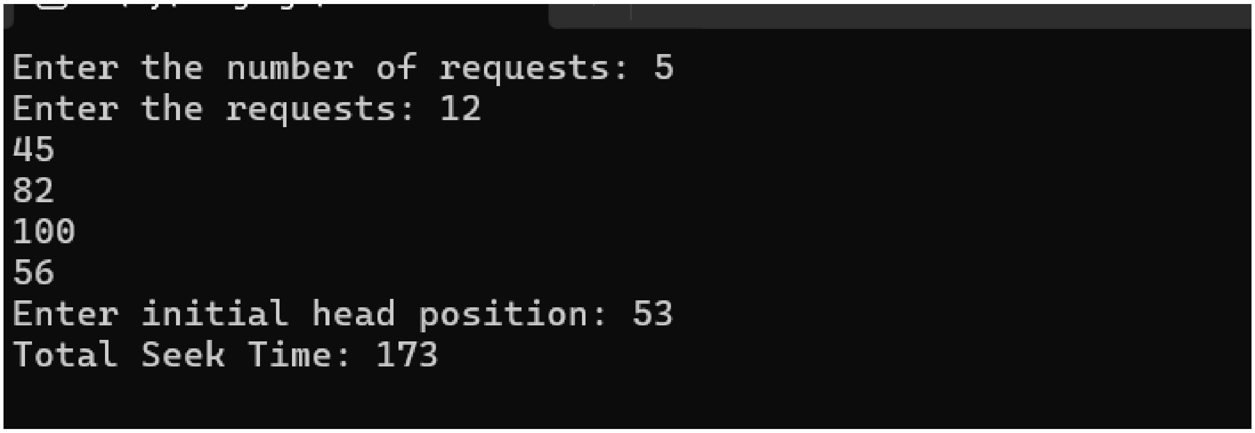


```
Enter the number of processes: 6
Enter the allocation for process 1: 2021
Enter the allocation for process 2: 0111
Enter the allocation for process 3: 4102
Enter the allocation for process 4: 1001
Enter the allocation for process 5: 1100
Enter the allocation for process 6: 1011
Enter the maximum for process 1: 9555
Enter the maximum for process 2: 2233
Enter the maximum for process 3: 7544
Enter the maximum for process 4: 3332
Enter the maximum for process 5: 5221
Enter the maximum for process 6: 4444
Enter the available resources: 3233
Need of process 1: 7534
The system is not in a safe state.
```

4

```
#include <stdio.h>

#include <stdlib.h>

int main() {

    int n, head, total_seek = 0;

    printf("Enter the number of requests: ");

    scanf("%d", &n);

    int requests[n];

    printf("Enter the requests: ");

    for (int i = 0; i < n; i++) {

        scanf("%d", &requests[i]);

    }

    printf("Enter initial head position: ");

    scanf("%d", &head);
    for (int i = 0; i < n; i++) {

        total_seek += abs(requests[i] - head);

        head = requests[i];

    }


    printf("Total Seek Time: %d\n", total_seek);

    return 0;

}
```

Output



```
Enter the number of requests: 5
Enter the requests: 12
45
82
100
56
Enter initial head position: 53
Total Seek Time: 173
```