

```

1 #include <stdio.h>
2
3 void worstFit(int blockSize[], int m, int processSize[], int n) {
4     int allocation[n];
5     for (int i = 0; i < n; i++) allocation[i] = -1;
6
7     for (int i = 0; i < n; i++) {
8         int worstIdx = -1;
9         for (int j = 0; j < m; j++) {
10             if (blockSize[j] >= processSize[i]) {
11                 if (worstIdx == -1 || blockSize[worstIdx] < blockSize[j]) {
12                     worstIdx = j;
13                 }
14             }
15         }
16         if (worstIdx != -1) {
17             allocation[i] = worstIdx;
18             blockSize[worstIdx] -= processSize[i];
19         }
20     }
21
22     printf("Process No.\tBlock No.\n");
23     for (int i = 0; i < n; i++) {
24         printf("%d\t\t", i + 1);
25         if (allocation[i] != -1) {
26             printf("%d\n", allocation[i] + 1);
27         } else {
28             printf("Not Allocated\n");
29         }
30     }
31 }
32
33 int main() {
34     int blockSize[] = {100, 500, 200, 300, 600};
35     int processSize[] = {212, 417, 112, 426};
36     int m = sizeof(blockSize) / sizeof(blockSize[0]);
37     int n = sizeof(processSize) / sizeof(processSize[0]);
38     worstFit(blockSize, m, processSize, n);
39     return 0;
40 }

```

^ System is in a safe state.

=== Code Execution Successful ===

```

1 #include <stdio.h>
2
3 void bestFit(int blockSize[], int m, int processSize[], int n) {
4     int allocation[n];
5     for (int i = 0; i < n; i++) allocation[i] = -1;
6
7     for (int i = 0; i < n; i++) {
8         int bestIdx = -1;
9         for (int j = 0; j < m; j++) {
10             if (blockSize[j] >= processSize[i]) {
11                 if (bestIdx == -1 || blockSize[bestIdx] > blockSize[j])
12                     bestIdx = j;
13             }
14         }
15         if (bestIdx != -1) {
16             allocation[i] = bestIdx;
17             blockSize[bestIdx] -= processSize[i];
18         }
19     }
20
21     printf("Process No.\tBlock No.\n");
22     for (int i = 0; i < n; i++) {
23         if (allocation[i] != -1)
24             printf("%d\t\t%d\n", i + 1, allocation[i] + 1);
25         else
26             printf("%d\t\tNot Allocated\n", i + 1);
27     }
28 }
29
30 int main() {
31     int blockSize[] = {100, 500, 200, 300, 600};
32     int processSize[] = {212, 417, 112, 426};
33     int m = sizeof(blockSize) / sizeof(blockSize[0]);
34     int n = sizeof(processSize) / sizeof(processSize[0]);
35     bestFit(blockSize, m, processSize, n);
36     return 0;
37 }

```

Process No. Block No.

1	4
2	2
3	3
4	5

=== Code Execution Successful