

programiz.com

sabari8883/operatin... Online C Compiler Online C Compiler Online C Compiler Write C programs to... two threads(semaph... Introducing ChatGPT Shared Memory Se...

Programiz
C Online Compiler

fiverr. *App development?*
We're up for it. **Find a freelancer**

Interactive C Course

main.c Run Output Clear

```
1 #include <stdio.h>
2
3 #define NUM_PARTITIONS 6
4 #define NUM_PROCESSES 5
5
6 int main() {
7     int partitions[NUM_PARTITIONS] = {300, 600, 350, 200, 750, 125};
8     int processes[NUM_PROCESSES] = {115, 500, 358, 200, 375};
9     int allocation[NUM_PROCESSES];
10    int i, j;
11
12    // Initialize allocation array to -1 (unallocated)
13    for (i = 0; i < NUM_PROCESSES; i++) {
14        allocation[i] = -1;
15    }
16
17    // Allocate memory to each process
18    for (i = 0; i < NUM_PROCESSES; i++) {
19        int best_index = -1;
20        for (j = 0; j < NUM_PARTITIONS; j++) {
21            if (partitions[j] >= processes[i] &
22                if (best_index == -1 || partitions[j] <
23                    partitions[best_index]) {
24                        best_index = j;
25                }
26            }
27            if (best_index != -1) {
```

/tmp/gUw86pWNPz.o
Process Size Partition
1 115 6
2 500 2
3 358 5
4 200 4
5 375 5

programiz.com

sabari8883/operatin...

Online C Compiler

Online C Compiler

Write C programs to...

two threads(semaph...

Introducing ChatGPT

Shared Memory Se...

Programiz
C Online Compiler

fiverr. App development?
We're up for it.
Find a freelancer

Interactive C Course

main.c

20 +
21 +
22 -
23
24
25
26
27 -
28
29
30
31
32
33
34
35 -
36
37 -
38
39 -
40
41
42
43
44
45 }
46

```
for (j = 0; j < NUM_PARTITIONS; j++) {  
    if (partitions[j] >= processes[i]) {  
        if (best_index == -1 || partitions[j] <  
            partitions[best_index]) {  
            best_index = j;  
        }  
    }  
}  
  
if (best_index != -1) {  
    allocation[i] = best_index;  
    partitions[best_index] -= processes[i];  
}  
  
// Print the allocation  
printf("Process\tSize\tPartition \n");  
for (i = 0; i < NUM_PROCESSES; i++) {  
    printf("%d\t %d\t", i+1, processes[i]);  
    if (allocation[i] != -1) {  
        printf("%d\n", allocation[i]+1);  
    } else {  
        printf("Not allocated\n");  
    }  
}  
  
return 0;  
}
```

Run

Output

/tmp/gUw86pWnpZ.o

Process	Size	Partition
1	115	6
2	500	2
3	358	5
4	200	4
5	375	5

Clear