

```
1  #include<stdio.h>
2  void firstFit(int blockSize[], int m, int processSize[], int n)
3  {
4      int i, j;
5      int allocation[n];
6      for(i = 0; i < n; i++)
7      {
8          allocation[i] = -1;
9      }
10     for (i = 0; i < n; i++)
11     {
12         for (j = 0; j < m; j++)
13         {
14             if (blockSize[j] >= processSize[i])
15             {
16                 allocation[i] = j;
17
18                 blockSize[j] -= processSize[i];
19
20                 break;
21             }
22         }
23     }
24 }
25
26 printf("\nProcess No.\tProcess Size\tBlock no.\n");
27 for (int i = 0; i < n; i++)
28 {
29     printf(" %i\t\t\t", i+1);
30     printf("%i\t\t\t\t", processSize[i]);
31     if (allocation[i] != -1)
32         printf("%i", allocation[i] + 1);
33     else
34         printf("Not Allocated");
35     printf("\n");
```

```
15 {
16     {
17         allocation[i] = j;
18
19         blockSize[j] -= processSize[i];
20
21         break;
22     }
23 }
24
25 }
26 printf("\nProcess No.\tProcess Size\tBlock no.\n");
27 for (int i = 0; i < n; i++)
28 {
29     printf("%i\t\t\t", i+1);
30     printf("%i\t\t\t", processSize[i]);
31     if (allocation[i] != -1)
32         printf("%i", allocation[i] + 1);
33     else
34         printf("Not Allocated");
35     printf("\n");
36 }
37 }
38 int main()
39 {
40     int m;
41     int n;
42     int blockSize[] = {105, 520, 300, 400, 680};
43     int processSize[] = {212, 417, 112, 426};
44     m = sizeof(blockSize) / sizeof(blockSize[0]);
45     n = sizeof(processSize) / sizeof(processSize[0]);
46     firstFit(blockSize, m, processSize, n);
47     return 0;
48 }
```

Process No.	Process Size	Block no.
1	212	2
2	417	5
3	112	2
4	426	Not Allocated

Process exited after 0.0466 seconds with return value 0
Press any key to continue . . .