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

#define MAX 100

int n, m, a[MAX];

void worst_fit(int block_size[], int process_size[])
{
   int allocation[n];
   int i, j;
   for (i = 0; i < n; i++)
   {
      int w = -1;
      for (j = 0; j < m; j++)
      {
         if (block_size[j] >= process_size[i])
         {
            if (w == -1)
               w = j;
            else if (block_size[w] < block_size[j])
               w = j;
         }
      }
      if (w != -1)
      {
         allocation[i] = w + 1;
         block_size[w] -= process_size[i];
      }
      else
      {
```

```
            allocation[i] = -1;
        }
    }
    printf("\nProcess No.\tProcess Size\tBlock no.\n");
    for (i = 0; i < n; i++)
    {
        printf("%d\t\t%d\t\t", i + 1, process_size[i]);
        if (allocation[i] != -1)
            printf("%d\n", allocation[i]);
        else
            printf("Not Allocated\n");
    }
}

int main()
{
    int block_size[MAX], process_size[MAX];
    int i;
    printf("Enter number of blocks: ");
    scanf("%d", &m);
    printf("Enter size of each block:\n");
    for (i = 0; i < m; i++)
    {
        printf("Block %d: ", i + 1);
        scanf("%d", &block_size[i]);
    }
    printf("Enter number of processes: ");
    scanf("%d", &n);
    printf("Enter size of each process:\n");
    for (i = 0; i < n; i++)
    {
```

```
        printf("Process %d: ", i + 1);
        scanf("%d", &process_size[i]);
    }
    worst_fit(block_size, process_size);
    return 0;
}
```

```
Enter number of blocks: 4
Enter size of each block:
Block 1: 100
Block 2: 500
Block 3: 200
Block 4: 300
Enter number of processes: 5
Enter size of each process:
Process 1: 50
Process 2: 150
Process 3: 300
Process 4: 200
Process 5: 100

Process No. Process Size    Block no.
1       50      4
2       150     2
3       300     2
4       200     3
5       100     1
```