```c
#include <stdio.h>
#include<conio.h>
int main() {
    int blockSize[10], processSize[10];
    int blockCount, processCount;
    int allocation[10]; // To store block index assigned to each process
    int i,j;
    // Input number of blocks and processes
    printf("Enter the number of memory blocks: ");
    scanf("%d", &blockCount);

    printf("Enter the number of processes: ");
    scanf("%d", &processCount);

    // Input sizes of memory blocks
    printf("Enter the sizes of memory blocks:\n");
    for (i=0; i<blockCount; i++)
    {
printf("Block %d: ", i + 1);
scanf("%d", &blockSize[i]);
    }

    // Input sizes of processes
    printf("Enter the sizes of processes:\n");
    for (i = 0; i < processCount; i++) {
printf("Process %d: ", i + 1);
scanf("%d", &processSize[i]);
    }

    // Initially no block is allocated
    for (i = 0; i < processCount; i++) {
allocation[i] = -1;
    }

    // Best Fit Allocation
    for (i = 0; i < processCount; i++) {
int bestIdx = -1;
for (j = 0; j < blockCount; j++) {
    if (blockSize[j] >= processSize[i]) {
 if (bestIdx == -1 || blockSize[j] < blockSize[bestIdx]) {
     bestIdx = j;
 }
    }
}

if (bestIdx != -1) {
    // Allocate block j to process i
    allocation[i] = bestIdx;
    blockSize[bestIdx] -= processSize[i];
}
    }

    // Display Allocation Results
    printf("\nProcess No.\tProcess Size\tBlock No.\n");
    for (i = 0; i < processCount; i++) {
```

```c
    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");
    }
    getch();
    return 0;
}
```