BEST FIT:

```
#include <stdio.h>

#define MAX_BLOCKS 10
#define MAX_PROCESSES 5

// Function to implement the Best Fit memory allocation algorithm
void bestFit(int blockSize[], int m, int processSize[], int n) {
    int allocation[n]; // To store the allocation of processes

    // Initially, no process is allocated any block
    for (int i = 0; i < n; i++) {
        allocation[i] = -1;
    }

    // Find best fit for each process
    for (int i = 0; i < n; i++) {
        int bestIdx = -1;
        for (int j = 0; j < m; j++) {
            // If the block is big enough and not yet allocated, and it's the best fit
            if (blockSize[j] >= processSize[i] && (bestIdx == -1 || blockSize[j] < blockSize[bestIdx])) {
                bestIdx = j;
            }
        }

        // If we found a suitable block
        if (bestIdx != -1) {
            allocation[i] = bestIdx; // Allocate the block to the process
            blockSize[bestIdx] -= processSize[i]; // Reduce the block size
        }
    }

    // Print the allocation result
    printf("\nProcess No.\tProcess Size\tBlock No.\tBlock Size\n");
    for (int i = 0; i < n; i++) {
        if (allocation[i] != -1) {
            printf("%d\t\t%d\t\t%d\t\t%d\n", i+1, processSize[i], allocation[i]+1,
blockSize[allocation[i]]);
        } else {
            printf("%d\t\t%d\t\tNot Allocated\n", i+1, processSize[i]);
        }
    }
}
```

```
int main() {
    int blockSize[MAX_BLOCKS], processSize[MAX_PROCESSES];
    int m, n;

    // Get the number of blocks and processes
    printf("Enter number of blocks: ");
    scanf("%d", &m);
    printf("Enter number of processes: ");
    scanf("%d", &n);

    // Get the block sizes
    printf("\nEnter block sizes:\n");
    for (int i = 0; i < m; i++) {
        printf("Block %d size: ", i+1);
        scanf("%d", &blockSize[i]);
    }

    // Get the process sizes
    printf("\nEnter process sizes:\n");
    for (int i = 0; i < n; i++) {
        printf("Process %d size: ", i+1);
        scanf("%d", &processSize[i]);
    }

    // Call bestFit to allocate blocks to processes
    bestFit(blockSize, m, processSize, n);

    return 0;
}
```

```
Enter number of blocks: 3
Enter number of processes: 3

Enter block sizes:
Block 1 size: 3
Block 2 size: 4
Block 3 size: 1

Enter process sizes:
Process 1 size: 6
Process 2 size: 4
Process 3 size: 2

Process No. Process Size    Block No.   Block Size
1       6       Not Allocated
2       4       2       0
3       2       1       1
```