BEST FIT

```
#include <stdio.h>
void bestFit(int blockSize[], int m, int processSize[], int n) {
    int allocation[n];
    for (int i = 0; i < n; i++) allocation[i] = -1;
    for (int i = 0; i < n; i++) {
        int bestIdx = -1;
        for (int j = 0; j < m; j++) {
            if (blockSize[j] >= processSize[i]) {
                if (bestIdx == -1 || blockSize[j] < blockSize[bestIdx]) {
                    bestIdx = j;
                }
            }
        }
        if (bestIdx != -1) {
            allocation[i] = bestIdx;
            blockSize[bestIdx] -= processSize[i];
        }
    }
    printf("\nBest Fit Allocation:\n");
    for (int i = 0; i < n; i++) {
        printf("Process %d (Size %d) -> ", i+1, processSize[i]);
        if (allocation[i] != -1)
            printf("Block %d\n", allocation[i]+1);
        else
            printf("Not Allocated\n");
    }
}
```

```
int main() {
    int m, n;
    printf("Enter number of memory blocks: ");
    scanf("%d", &m);
    int blockSize[m];
    printf("Enter sizes of %d memory blocks:\n", m);
    for (int i = 0; i < m; i++) scanf("%d", &blockSize[i]);
    printf("Enter number of processes: ");
    scanf("%d", &n);
    int processSize[n];
    printf("Enter sizes of %d processes:\n", n);
    for (int i = 0; i < n; i++) scanf("%d", &processSize[i]);
    bestFit(blockSize, m, processSize, n);
    return 0;
}
```