Program:

```
import java.util.Scanner;
public class BestFit {
        static void bestFit(int blockSize[], int m, int processSize[],int n,int remblockSize[])
        {
                int allocation[] = new int[n];
                for (int i = 0; i < allocation.length; 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)
                                                bestIdx = j;
                                        else if (blockSize[bestIdx] > blockSize[j])
                                                bestIdx = j;
                                }}
                        if (bestIdx != -1)
                        {
                                allocation[i] = bestIdx;
                                blockSize[bestIdx] -= processSize[i];
                                remblockSize[i]=blockSize[bestIdx];
                        }}
    System.out.println("\nProcess No.\tProcess Size\tBlock no.\tRemaninig Block Size");
                        for (int i = 0; i < n; i++)
                        {
                                System.out.print(" " + (i+1) + "\t\t" + processSize[i] + "\t\t");
                                if (allocation[i] != -1) {
                                        System.out.print((allocation[i] + 1)+"\t\t"+remblockSize[i]);  }
                                else {
                                        System.out.print("Not Allocated"+"\t"+remblockSize[i]);  }
                                System.out.println();}}
    public static void main(String[] args) {
```

```
int m,n,num;
Scanner in=new Scanner(System.in);
System.out.print("Enter how many number of blocks you want to enter:");
m=in.nextInt();
int remblockSize[]=new int[m];
int blockSize[]=new int[m];
for(int i=0;i<m;i++) {
        System.out.print("Enter Data "+(i+1)+":");
        num=in.nextInt();
        blockSize[i]=num;
}
System.out.print("Enter how many number of process you want to enter:");
n=in.nextInt();
int processSize[]=new int[n];
for(int i=0;i<n;i++) {
        System.out.print("Enter Data "+(i+1)+":");
        num=in.nextInt();
        processSize[i]=num;
}
bestFit(blockSize, m, processSize, n,remblockSize); }}
```

Output:

```
Enter how many number of blocks you want to enter:4
Enter Data 1:10
Enter Data 2:15
Enter Data 3:15
Enter Data 4:15
Enter how many number of process you want to enter:4
Enter Data 1:10
Enter Data 2:15
Enter Data 3:14
Enter Data 4:16
```

Process No.	Process Size	Block no.	Remaninig Block Size
1	10	1	0
2	15	2	0
3	14	3	1
4	16	Not Allocated	0