NAME: G. GANESH

REG NO: 192373008

EXERICSE-22

Construct a C program to implement the best fit algorithm of memory management.

Aim:

To construct a C program to implement the Best Fit memory allocation algorithm.

Algorithm:

- **Initialize Memory Blocks and Processes:**
    - Create arrays for memory block sizes and process sizes.
- **Find Best Fit:**
    - For each process, find the memory block with the smallest size that can accommodate it.
    - Allocate the process to this block and reduce the block size.
- **Unallocated Processes:**
    - If no suitable block is found, mark the process as unallocated.
- **Display Results:**
    - Show the allocation details for each process.

Procedure:

1. Input the sizes of memory blocks and processes.
2. For each process, iterate through all memory blocks to find the best fit.
3. Allocate the process to the best-fit block and adjust the block size.
4. Print the allocation details for each process and memory block.

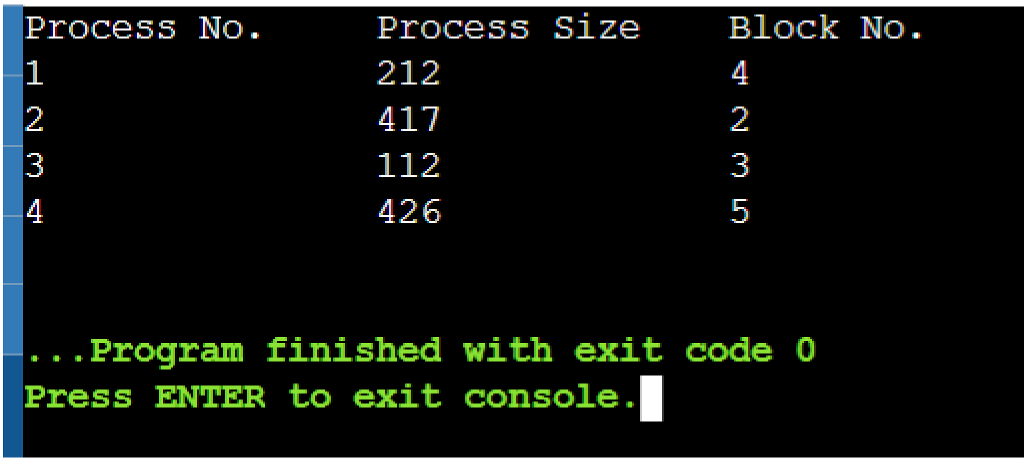
Code:

```
#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; // Initialize allocation array

    }

    for (int i = 0; i < n; i++) {

        int bestIndex = -1;

        for (int j = 0; j < m; j++) {
            if (blockSize[j] >= processSize[i]) {
                if (bestIndex == -1 || blockSize[j] < blockSize[bestIndex]) {
                    bestIndex = j;
                }
            }
        }
        if (bestIndex != -1) {
            allocation[i] = bestIndex;
            blockSize[bestIndex] -= processSize[i];
        }
    }
    printf("Process No.\tProcess Size\tBlock No.\n");
    for (int i = 0; i < n; i++) {
        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");
        }
    }
}
int main() {
    int blockSize[] = {100, 500, 200, 300, 600};
    int processSize[] = {212, 417, 112, 426};
    int m = sizeof(blockSize) / sizeof(blockSize[0]);
    int n = sizeof(processSize) / sizeof(processSize[0]);
    bestFit(blockSize, m, processSize, n);
    return 0;
}
```

Result:

The program successfully implements the Best Fit memory allocation algorithm, assigning processes to the memory blocks with the smallest sufficient available space.

Output:



```
Process No.        Process Size     Block No.
1                  212              4
2                  417              2
3                  112              3
4                  426              5


...Program finished with exit code 0
Press ENTER to exit console.
```