Code:

```
// Write programs to simulate the Best Fit Memory Allocation Technique.

#include <bits/stdc++.h>
using namespace std;

int main() {

  int num_blocks, num_processes;
  cout << "Enter the number of memory blocks: ";
  cin >> num_blocks;

  cout << "\nEnter the number of processes: ";
  cin >> num_processes;

  int mem_blocks[num_blocks];
  cout << "\nEnter the sizes of the memory blocks(in K): " << endl;
  for (int i = 0; i < num_blocks; i++) {
    cin >> mem_blocks[i];
  }

  int process_sizes[num_processes];
  cout << "\nEnter Sizes of the processes(in K): " << endl;
  for (int i = 0; i < num_processes; i++) {
    cin >> process_sizes[i];
  }

  bool allocation_status[num_processes];
  memset(allocation_status, false, sizeof(allocation_status));

  for (int i = 0; i < num_processes; i++) {
    int process_size = process_sizes[i];
    int best_fit_index = -1;
    int smallest_fit = INT_MAX;

    for (int j = 0; j < num_blocks; j++) {
      if (mem_blocks[j] >= process_size && mem_blocks[j] < smallest_fit) {
        best_fit_index = j;
        smallest_fit = mem_blocks[j];
      }
    }
    if (best_fit_index != -1) {
      allocation_status[i] = true;
      mem_blocks[best_fit_index] -= process_size;
    } else {
      cout << "Could not allocate memory for " << process_sizes[i] << "K"
           << endl;
```

```
    }
  }

  for (int i = 0; i < num_processes; i++) {
    if (allocation_status[i]) {
      cout << process_sizes[i] << "K has been allocated memory" << endl;
    } else {
      cout << process_sizes[i] << "K could not be allocated memory" << endl;
    }
  }

  return 0;
}
```

Sample Output:

```
Enter the number of memory blocks: 5

Enter the number of processes: 4

Enter the sizes of the memory blocks(in K):
100 500 200 300 600

Enter Sizes of the processes(in K):
212 417 112 426

212K has been allocated memory
417K has been allocated memory
112K has been allocated memory
426K has been allocated memory
```