Code:

// Write programs to simulate the Worst Fit Memory Allocation Technique.

```cpp
#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 << "Enter the number of processes: ";
  cin >> num_processes;

  int memory_blocks[num_blocks];
  cout << "Enter the sizes of the memory blocks:" << endl;
  for (int i = 0; i < num_blocks; i++) {
    cin >> memory_blocks[i];
  }

  int process_sizes[num_processes];
  cout << "Enter sizes of the processes:" << 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 worst_fit_index = -1;
    int largest_fit = -1;
    for (int j = 0; j < num_blocks; j++) {
      if (memory_blocks[j] >= process_size && memory_blocks[j] > largest_fit) {
        worst_fit_index = j;
        largest_fit = memory_blocks[j];
      }
    }
    if (worst_fit_index != -1) {
      allocation_status[i] = true;
      memory_blocks[worst_fit_index] -= process_size;
    } else {
      cout << "Could not allocate memory for " << process_sizes[i] << "K"
           << endl;
    }
  }
```

```cpp
  // Print the allocation status of each process
  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:
150 500 200 300 550

Enter sizes of the processes:
220 430 110 425

Could not allocate memory for 425K
220K has been allocated memory
430K has been allocated memory
110K has been allocated memory
425K could not be allocated memory