```
#include <stdio.h>
#define MEMORY SIZE 1000
#define NUM_BLOCKS 5
int memory[MEMORY_SIZE];
int block_size[NUM_BLOCKS] = {100, 200, 300, 150, 250};
int block_allocated[NUM_BLOCKS] = {0};
void allocate_memory(int process_size, int process_id) {
  for (int i = 0; i < NUM_BLOCKS; i++) {</pre>
    if (!block_allocated[i] && block_size[i] >= process_size) {
      block_allocated[i] = process_id;
      printf("Process %d allocated to block %d\n", process_id, i);
      return;
    }
  }
  printf("Process %d cannot be allocated\n", process_id);
void deallocate_memory(int process_id) {
  for (int i = \overline{0}; i < NUM BLOCKS; \overline{i}++) {
    if (block allocated[i] == process id) {
      block allocated[i] = 0;
      printf("Process %d deallocated from block %d\n", process id, i);
      return;
    }
  }
  printf("Process %d not found in memory\n", process id);
int main() {
  allocate memory(120, 1);
  allocate memory(180, 2);
  allocate memory(90, 3);
  allocate memory(200, 4);
  allocate memory(80, 5);
  deallocate memory(2);
  allocate memory(180, 6);
  return 0;
```