```c
#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) {
  int best_index = -1;
  for (int i = 0; i < NUM_BLOCKS; i++) {
    if (!block_allocated[i] && block_size[i] >= process_size) {
      if (best_index == -1 || block_size[i] < block_size[best_index]) {
        best_index = i;
      }
    }
  }

  if (best_index != -1) {
    block_allocated[best_index] = process_id;
    printf("Process %d allocated to block %d\n", process_id, best_index);
  } else {
    printf("Process %d cannot be allocated\n", process_id);
  }
}

void deallocate_memory(int process_id) {
  for (int i = 0; i < NUM_BLOCKS; 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;
}
```