```
#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 worst_index = -1;
  for (int \bar{i} = 0; i < NUM_BLOCKS; i++) {
    if (!block_allocated[i] && block_size[i] >= process_size) {
      if (worst_index == -1 || block_size[i] > block_size[worst_index]) {
        worst index = i;
      }
   }
  }
  if (worst_index != -1) {
    block allocated[worst index] = process id;
    printf("Process %d allocated to block %d\n", process id, worst index);
  } else {
    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;
```