```
#include <stdio.h>

#define MAX_MEMORY_BLOCKS 100
#define INITIAL_MEMORY_SIZE 1000

int memory_blocks[MAX_MEMORY_BLOCKS]; // Array to store memory block sizes
int num_blocks = 0; // Number of memory blocks

// Initialize memory with the initial size
void initialize_memory() {
    memory_blocks[0] = INITIAL_MEMORY_SIZE;
    num_blocks = 1;
}

// Display the memory blocks with sizes
void display_memory() {
    printf("Memory Blocks:\n");
    for (int i = 0; i < num_blocks; i++) {
        printf("Block %d: %d KB\n", i + 1, memory_blocks[i]);
    }
    printf("\n");
}

// Allocate memory using worst fit algorithm
void allocate_worst_fit(int size) {
    int worst_fit_index = -1;
    int largest_block_size = 0;

    for (int i = 0; i < num_blocks; i++) {
        if (memory_blocks[i] >= size && memory_blocks[i] > largest_block_size) {
            largest_block_size = memory_blocks[i];
```

```
            worst_fit_index = i;
        }
    }

    if (worst_fit_index == -1) {
        printf("Memory allocation failed. Not enough contiguous memory available.\n");
    } else {
        memory_blocks[worst_fit_index] -= size;
        // Insert new block if there is remaining memory
        if (memory_blocks[worst_fit_index] > 0) {
            for (int i = num_blocks; i > worst_fit_index + 1; i--) {
                memory_blocks[i] = memory_blocks[i - 1];
            }
            memory_blocks[worst_fit_index + 1] = memory_blocks[worst_fit_index];
            num_blocks++;
        }
        printf("Memory allocated successfully: %d KB\n", size);
    }
}

int main() {
    initialize_memory();

    display_memory();

    allocate_worst_fit(200);
    display_memory();

    allocate_worst_fit(500);
    display_memory();
```

```
    allocate_worst_fit(800);
    display_memory();

    allocate_worst_fit(300);
    display_memory();

    return 0;
}
```