```c
#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 best fit algorithm
void allocate_best_fit(int size) {
    int best_fit_index = -1;
    int smallest_block_size = INITIAL_MEMORY_SIZE + 1;

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

```c
                best_fit_index = i;
            }
        }


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

int main() {
    initialize_memory();


    display_memory();


    allocate_best_fit(200);
    display_memory();


    allocate_best_fit(500);
    display_memory();
```

```
    allocate_best_fit(800);

    display_memory();


    allocate_best_fit(300);

    display_memory();


    return 0;
}
```
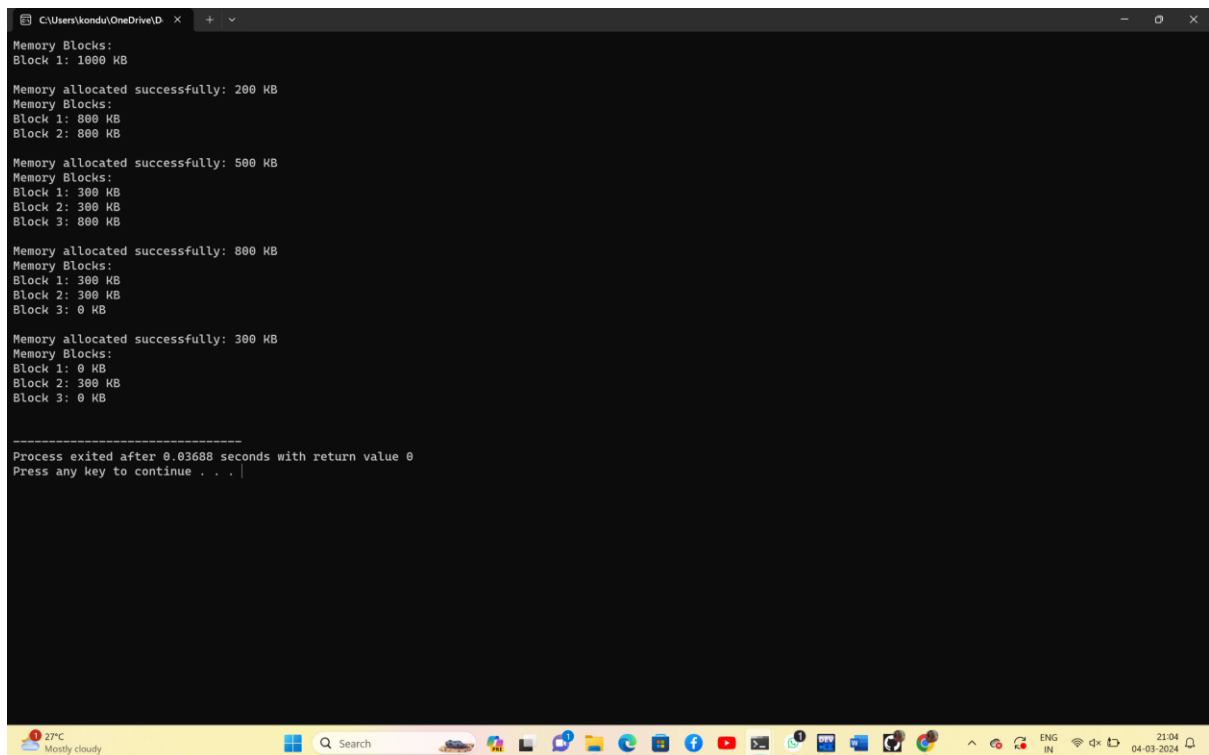


```
Memory Blocks:
Block 1: 1000 KB

Memory allocated successfully: 200 KB
Memory Blocks:
Block 1: 800 KB
Block 2: 800 KB

Memory allocated successfully: 500 KB
Memory Blocks:
Block 1: 300 KB
Block 2: 300 KB
Block 3: 800 KB

Memory allocated successfully: 800 KB
Memory Blocks:
Block 1: 300 KB
Block 2: 300 KB
Block 3: 0 KB

Memory allocated successfully: 300 KB
Memory Blocks:
Block 1: 0 KB
Block 2: 300 KB
Block 3: 0 KB


--------------------------------
Process exited after 0.03688 seconds with return value 0
Press any key to continue . . .
```