DESIGN DOCUMENT

* Files:
    - heap_management.c
    - heap_management.h
    - allocation_deallocation.c
    - my_memory_heap.c
    - makefile

* Heap_management.c contains the main function alongwith the initialization and demonstration function. This program demonstrates all the test cases for malloc, calloc, realloc, free, defragmentation.

* Heap_management.h contains all the pre-processor directives, global variables and macros, function prototypes.

* my_memory_heap.c contains functions which deals with the static arrays which is taken as a Heap. It has Functions like defragmentation function, finding free space in heap, copy data from one block of heap memory to another block of heap memory, fusion function which merges to the two blocks, valid address function to find that a address was malloc'ed in our heap or not, find_block function to find a free block whenever a demand is made. Extend_heap will get more memory for the user whenever required, split_block function will split a block into two if it is too large than the required size.

* allocation_deallocation.c contains 4 functions – my_realloc, my_malloc, free, my_calloc.

* Code files are heavily commented and are very easily readable to understand the flow of program and the approach taken.

* DATA STRUCTURE:

```
// Global Variables
char *end_limit;                    // Last Address of Heap
char *limit;                        // Break Position of Heap
char memory[MEM_SIZE];              // Memory or Heap
extern void *base;                  // Base Address of Heap Memory
struct block *traverse;             // To Traverse to End

// Structure of Meta-data
struct block
{
```

```
        size_t size;                    // Size of Space Allocated to a Block
        struct block *next;             // Next Block Address
        struct block *prev;             // Previous Block Address
        int free;                       // Free(1) or Occupied(0)
};

// Function Prototype
void* my_malloc(size_t);                // Malloc Function
void* my_calloc(size_t, size_t);        // Calloc Function
void init();                            // Initialization Function
void* my_sbrk(size_t);                  // sbrk Function
void demo_malloc();                     // Demonstrating Malloc
struct block* find_block( size_t);      // Find Free Block
struct block* extend_heap( size_t);     // Extend Heap
void split_block( struct block*, size_t); // Split Block
struct block* fusion(struct block *);   // Fusion of Two Free Blocks
int valid( void*);                      // Valid Function
void free( void*);                      // Free Memory
void* my_realloc( void *, size_t);      // Realloc Function
void copy_data (void *, void *);        // Copy Data
size_t free_space();                    // Free Space
void defragment_my_heap();              // Defragment Heap
```