Problem 4.c

```c

/**
 * Implementation of a Binary Heap (Max-Heap)
 * Author: Mithusayel Murmu
 */

#include <stdio.h>
#include <stdlib.h>

static inline void swap_data(int *a, int *b) {
    int t = *a; *a = *b; *b = t;
}

static inline int comp_asc(int a, int b) { return a - b; }

typedef int (*compare_func)(int, int);

/**
 * Recursively max-heapify a node at index idx
 * @arr_base: The integer array to use
 * @idx:      Index of node to max-heapify
 * @hsz:      Heap size
 * @compare:  Function to be used for comparison
 */
void max_heapify(int *arr_base, size_t idx, int hsz, compare_func comp) {
    int lt = 2 * idx + 1;
    int rt = 2 * (idx + 1);
    int max;

    if (lt < hsz && comp(arr_base[idx], arr_base[lt]) < 0)
        max = lt;
    else
        max = idx;
    if (rt < hsz && comp(arr_base[max], arr_base[rt]) < 0)
        max = rt;

    if (idx != max) {
        swap_data(arr_base + idx, arr_base + max);
        max_heapify(arr_base, max, hsz, comp);
    }
}

/**
 * Builds a max-heap out of the given array
 * @arr_base: The integer array to use
 * @asz:      Size of the array
 * @comp:     Function to be used for comparison
 */
void build_max_heap(int *arr_base, size_t asz, compare_func comp) {
    int i;
    for (i = asz / 2 - 1; i >= 0; i--)
        max_heapify(arr_base, i, asz, comp);
}

/**
 * Sorts an array of integers (Used as test for the Heap DS)
 * @arr_base: The integer array to sort
 * @asz:      Size of the array
 * @comp:     Function to be used for comparison
 */
void heap_sort(int *arr_base, size_t asz, compare_func comp) {
    if (asz < 2) return;

    int i;
    build_max_heap(arr_base, asz, comp);
    for (i = asz - 1; i > 0; i--) {
        swap_data(arr_base, arr_base + i);
        max_heapify(arr_base, 0, --asz, comp);
    }
}

/** Driver function */
int main(int argc, char const *argv[]) {
    int N, i = 0;

    printf("Number of integers to use: ");
    scanf("%d", &N);
    printf("Enter %d space separated integers: ", N);

```

```c
    int *arr = (int *) malloc(N * sizeof(int));
    while (N--)
        scanf("%d", &arr[i++]);

    heap_sort(arr, i, comp_asc);
    printf("\nAfter Heap Sort:\n");
    for (N = 0; N < i; N++)
        printf("%d ", arr[N]);
    printf("\n"); free(arr);

    return 0;
}

```