Homework #7

a) **(6 points)** Implement a function called *outputSorted* that takes an unsorted array of *Person* objects and outputs them to stdout in sorted order *using only a Heap.* The function should have this prototype:

```
void outputSorted (const Person people [ ],
        int numPeople,
        int (* compare) (const void *pKey1, const void *pKey2));
```

outputSorted should initialize a Heap with the given compare function, insert the people into the heap, then extract and output each person from the heap. The person array should not be modified by outputSorted.

Type Person should be defined as follows:

```
Typedef struct Person_ {
        const char *name;
        int age;
        double height;
} Person;
```

hw7 > C hw7.c > main()

```c
#include "heap.h"
#include <stdio.h>
#include <string.h>

void outputSorted (const Person people [], int numPeople,
                   int (*compare) (const void *pKey1, const void *pKey2)) {

    // a1. initialize a Heap with the given compare function
    Heap heap;
    heap_init(&heap, compare, NULL);

    // a2. insert the people into the heap
    for (int i=0; i < numPeople; i++) {
        heap_insert(&heap, &people[i]);
    }

    // a3. extract and output each person from the heap
    Person *currentPerson;
    while (heap_size(&heap) > 0) {
        heap_extract(&heap, (void**)&currentPerson);
        printf("Name: %s, Age: %d, Height: %.2fm\n",
               currentPerson->name,
               currentPerson->age,
               currentPerson->height);
    }

    heap_destroy(&heap);
}
```

b) **(1 point)** Demonstrate outputSorted taking an array of at least 5 unsorted people then outputting those people sorted by **ascending name.**

```c
int main() {
    Person people[] = {
        {"Jeff", 23, 1.73},
        {"Sean", 23, 1.75},
        {"Hugh", 61, 1.78},
        {"Joann", 67, 1.60},
        {"Yuka", 21, 1.50}
    };

    outputSorted(people, 5, compareName);
    return 0;
}
```

```c
int compareName(const void *pKey1, const void *pKey2) {
    const Person *p1 = (const Person *)pKey1;
    const Person *p2 = (const Person *)pKey2;

    // Sort by name
    return strcmp(p2->name, p1->name);
}
```

```
~/Desktop/DSA/hw7 main* ❯ ./hw7
Name: Hugh, Age: 61, Height: 1.78m
Name: Jeff, Age: 23, Height: 1.73m
Name: Joann, Age: 67, Height: 1.60m
Name: Sean, Age: 23, Height: 1.75m
Name: Yuka, Age: 21, Height: 1.50m
```

c) **(1 point)** Demonstrate outputSorted taking an array of at least 5 unsorted people then outputting those people sorted by **ascending age**.

```c
int compareAge(const void *pKey1, const void *pKey2) {
    const Person *p1 = (const Person *)pKey1;
    const Person *p2 = (const Person *)pKey2;

    // Sort by age
    return p2->age - p1->age;
}
```

```
~/Desktop/DSA/hw7 main* ❯ ./hw7
Name: Yuka, Age: 21, Height: 1.50m
Name: Sean, Age: 23, Height: 1.75m
Name: Jeff, Age: 23, Height: 1.73m
Name: Hugh, Age: 61, Height: 1.78m
Name: Joann, Age: 67, Height: 1.60m
```

d) **(1 point)** Demonstrate outputSorted taking an array of at least 5 unsorted people then outputting those people sorted by **ascending height**.

```c
int compareHeight(const void *pKey1, const void *pKey2) {
    const Person *p1 = (const Person *)pKey1;
    const Person *p2 = (const Person *)pKey2;

    // Sort by height (precision issues when subtracting doubles with each other)
    if (p2->height > p1->height) return 1;
    if (p2->height < p1->height) return -1;
    return 0;
}
```

```
~/Desktop/DSA/hw7 main* ❯ clang -o hw7 hw7.c heap.c
~/Desktop/DSA/hw7 main* ❯ ./hw7
Name: Yuka, Age: 21, Height: 1.50m
Name: Joann, Age: 67, Height: 1.60m
Name: Jeff, Age: 23, Height: 1.73m
Name: Sean, Age: 23, Height: 1.75m
Name: Hugh, Age: 61, Height: 1.78m
~/Desktop/DSA/hw7 main* ❯
```