I am at the point to where I am trying to code the consolidation function, although I have not gotten very far into it. I have calculated the size of the array. What I believe I am having issues on is initializing the array to NULL.

First, to make sure I am not doing something silly, is the array supposed to be an array of binheaps, such as:

```c
binheap *arr[arr_size];
```

or am I completely wrong?

Second, if I am allocating the array correctly, how do I go about initializing all the elements in the array to NULL?

Every time I try to access any index in the array, I get a segmentation fault, even when trying to just set a value for a single index:

```c
arr[0] = NULL;
```

---

My consolidation array D is an array of nodes, which I am allocating dynamically:

```c
def D;
    Dsize = ???; //calculate size of D array
    D = malloc(sizeof(node *) * Dsize);
    if (D == 0) Fatal("out of memory
    ");
    I initialize it with a loop:
    for (i = 0; i < Dsize; ++i) D[i] = 0;
```

---

allocating the array dynamically sound like a consistent approach to avoid fault, just like every c concept in this class. What are some bad effects if any from using the following code to initialize the array.

```c
memset(array,0,sizeof (array)); "taken from a web side".
```
If the array is allocated via malloc, then sizeof(array) is the size of a pointer, not the amount of memory allocated.

Otherwise, this would only work if a null pointer is represented with all zero bits. I don't know if that is guaranteed by the C standard.