

Enhanced malloc(), free() PA3 Silvia Carbajal

My header file contains the struct i will use for my project

The struct has a size (size of the space after the struct), is free(to see if the current struct was already freed), and 2 pointers for a struct

2 pointers because I'm creating a link of structs in an array so to make it easier for me i have a prev and such as the 2 pointers so i can traverse the link easily

My malloc function's design is based on a 50000 sized array

so basically it should look like this when malloc is used in the array

```
-----  
prev|succ|isfree|size|chunk to size|prev|succ|isfree|size|chunk to size| and so on  
-----
```

I divide my array into two different sections , the smaller section for allocations of size 100 or less and the larger section for more than 100 in size

40000 for the larger space and 10000 for the smaller space for fragmentation

and i have two roots one for each space

when some thing is allocated i give them a pointer to the beginning of the size they asked so after the struct of that chunk and then I create another struct after the allocated size

so its ready for the next one. I also have a check to see if the struct that has been freed has enough space for the next thing that is trying to be allocated so i don't create a new link and waste space.

my free function was faster to write because myMalloc function gave me the idea of how it all works. To avoid a segmentation fault , when a user tried to free a pointer that was not returned by myMalloc , i created a dummy struct pointer that went through the whole array and checked if it (dummy+struct size , which would give me what was returned by malloc)was equal to the pointer that the user was trying to free

I also had a check to see if it was already free and null just in case.

I had two pointers of the struct called pred and succ (for predecessor and successor)

I had a check to see if pred was free and if succ was free so I can unlink current pointer that is pointing to the currently freed struct and link it to either pred or succ or both.

That way there isn't 2 freed structs next to each other. Connecting the freed chunks increases the size of what was already freed.