

`realloc(ptr, new-size)`

expand the block for `ptr`  
to the new size if possible.

Returns `ptr` if it could or  
a newly-allocated block (elsewhere)  
if not, and frees `ptr`

`a = malloc(16)`

`b = malloc(16)`

`c = malloc(32)`

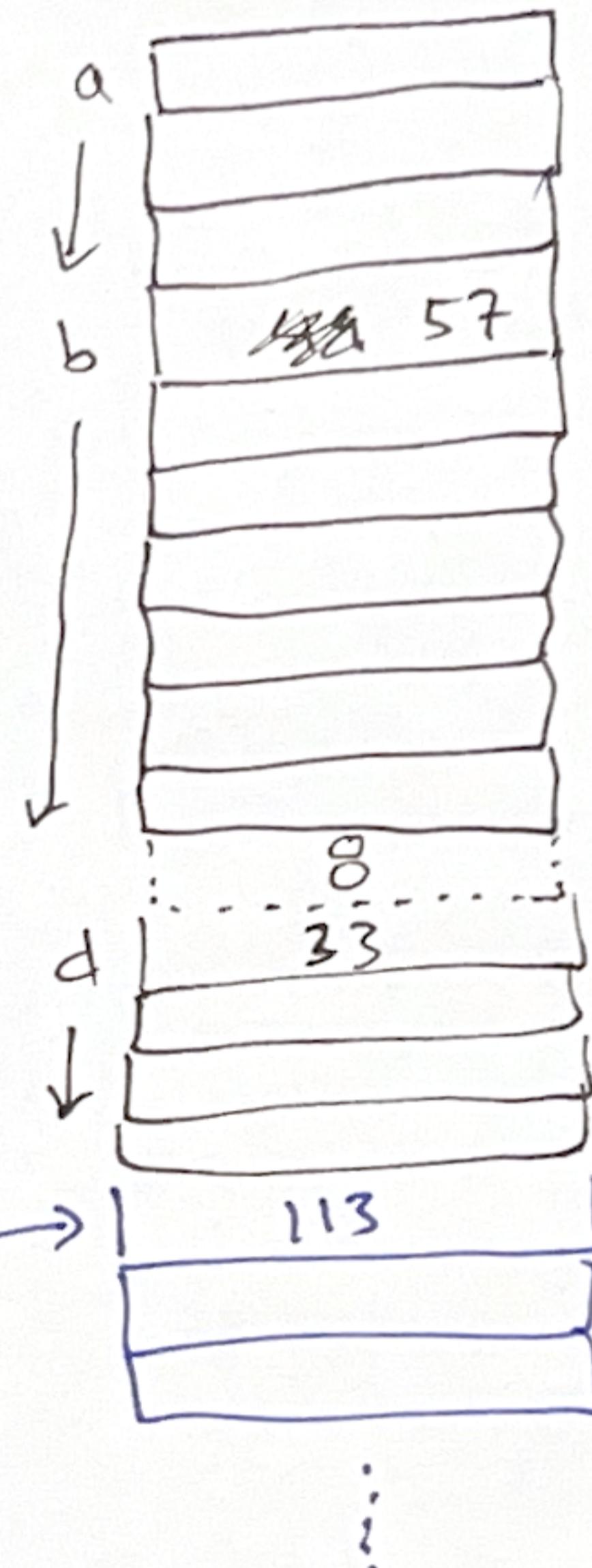
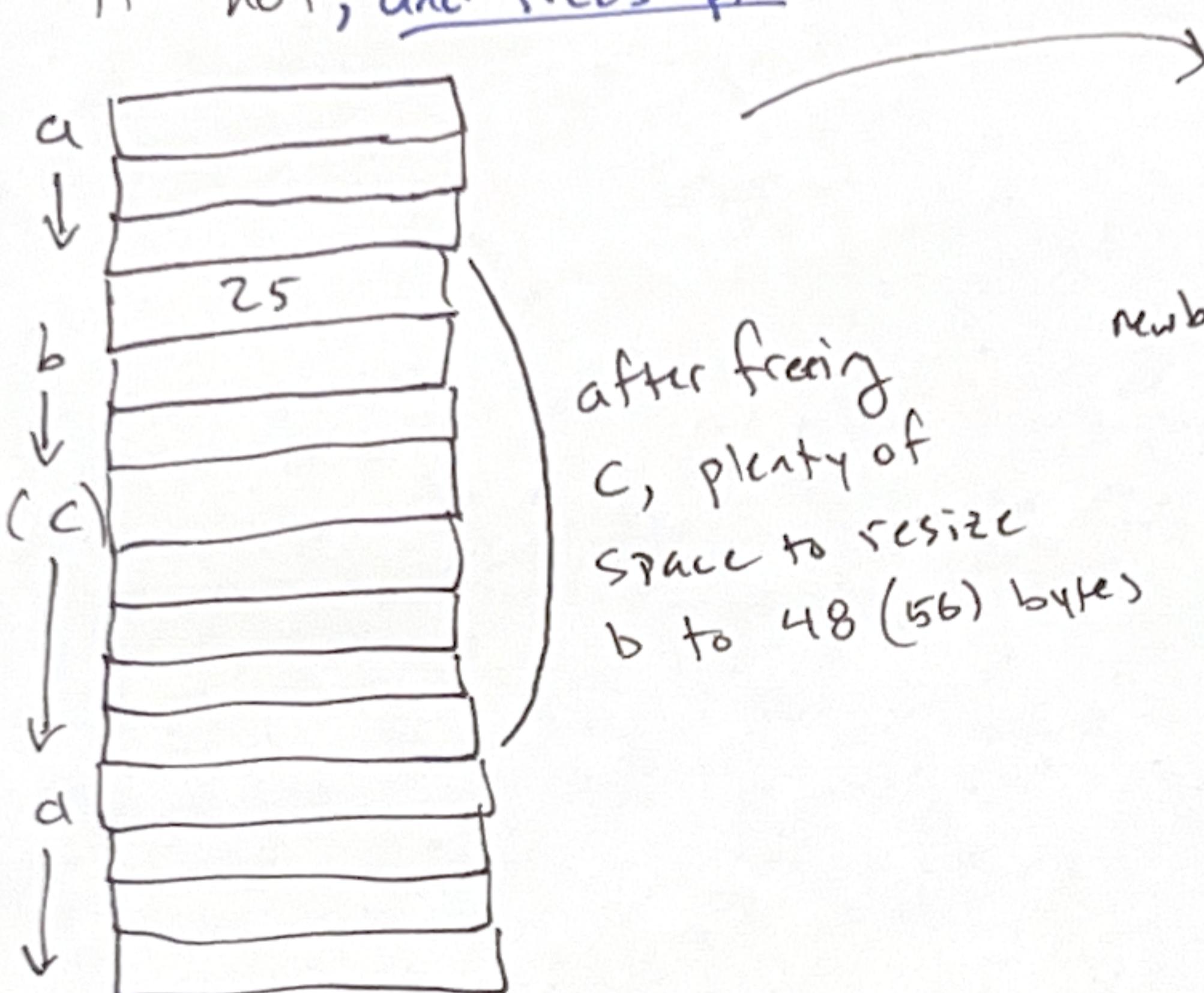
`d = malloc(24)`

`free(c)`

`newb = realloc(b, 48)`

`newb2 = realloc(b, 100)`

`Newb == b`



```
char* buffer = malloc(100);
while(...) {
    farts(buffer + offset ....)
    if(...) {
        buffer = realloc(buffer,
                          bigger_size)
    }
}
```

`void* calloc(size_t count, size_t size)`

$\approx$  `malloc(count * size)`

and it sets the memory to all 0's guaranteed

```

int vals[] = {1, 2, 3};
int pid = fork();
if(pid) {
    printf("%op", &vals);
    wait(NULL);
    printf ("%od", vals[0]);
}
else {
    printf("%op", &vals);
    vals[0] = 9999;
    printf ("%od", vals[0]);
}

```

4 Kib default page size

Virtual Memory

