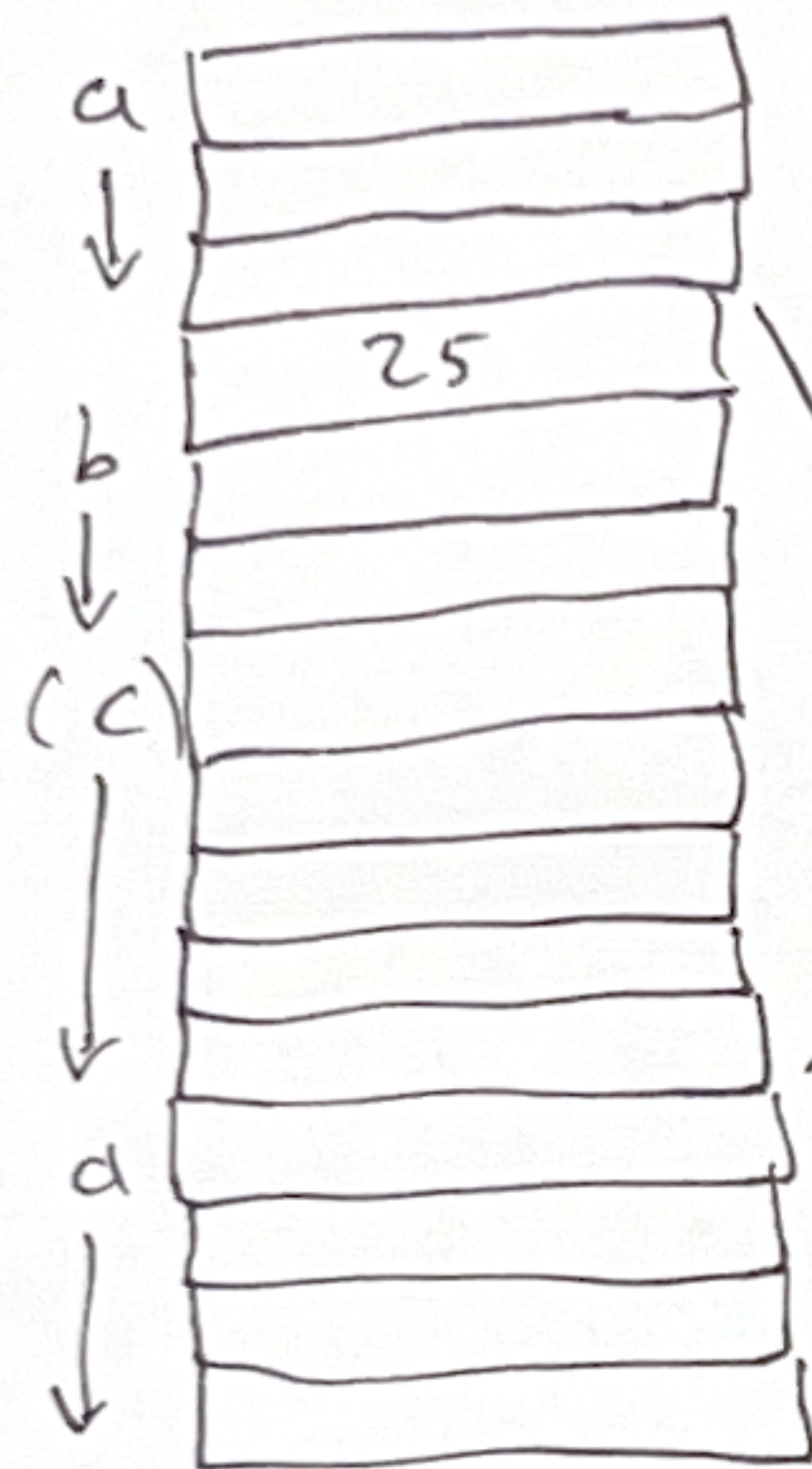


realloc(ptr, new-size)

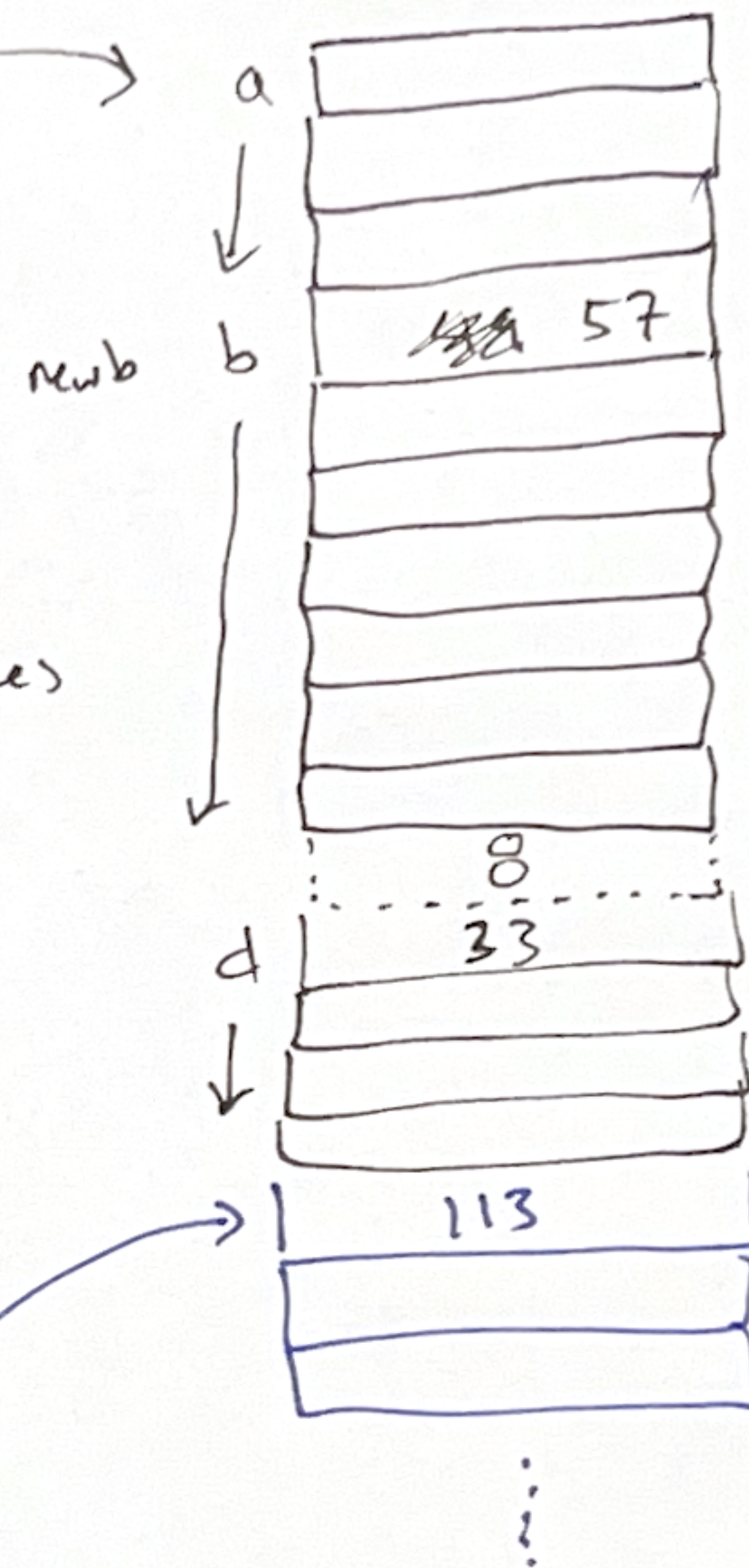
expand the block for ptr
to the new size if possible.
Returns ptr if it could or
a newly-allocated block (elsewhere)
if not, and frees ptr

```
a = malloc(16)
b = malloc(16)
c = malloc(32)
d = malloc(24)
free(c)
newb = realloc(b, 48)
newb2 = realloc(b, 100)
```

newb == b



after freeing
c, plenty of
space to resize
b to 48 (56) bytes



```
char* buffer = malloc(100);
while(...) {
    fgets(buffer + offset, ..., ...)
    if(...) {
        buffer = realloc(buffer,
                          bigger-size)
    }
}
```

void* calloc(size_t count, size_t size)

≈ malloc(count * size)

and it sets the memory to all 0's guaranteed

```

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

```

4 Kib default page size

Virtual Memory

0x

