Test

Violations

Memory Leak

Potentially run out of memory

not really a violations

Dangling Pointer

Pointer pointing to previously free'd memory

Dereference NULL

To avoid check for null before dereference

Link is now up. for HW1

For allocating memory we have malloc, calloc, realloc

previously discussed

For Recovering memory we use free

calloc is IMO safer

sets all bytes to zero

calloc (# items desired, size of each item);

↑ size_t

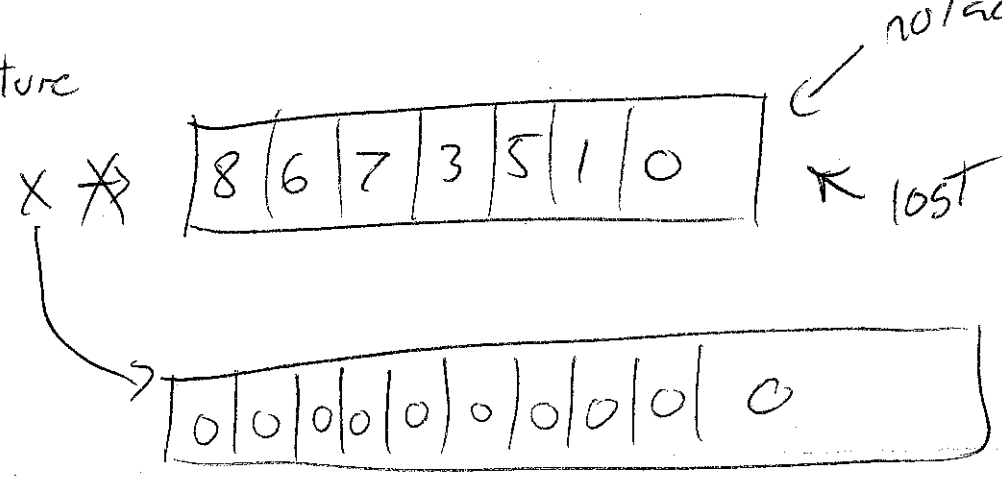
Seven integers

(int *) calloc (7, sizeof (int));

x = (int *) calloc (7, sizeof(int)); ←

x = (int *) calloc (10, sizeof(int)); beta

Picture

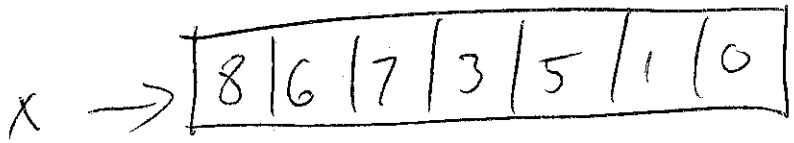


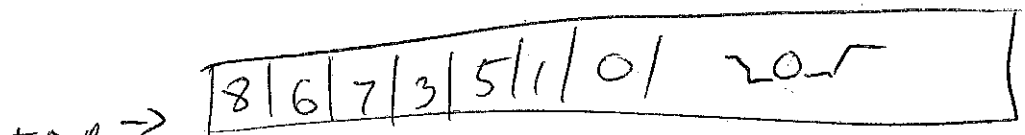
realloc

realloc takes in the reference to the old memory and the size the memory should become

realloc makes a new block of memory and copies in the old data

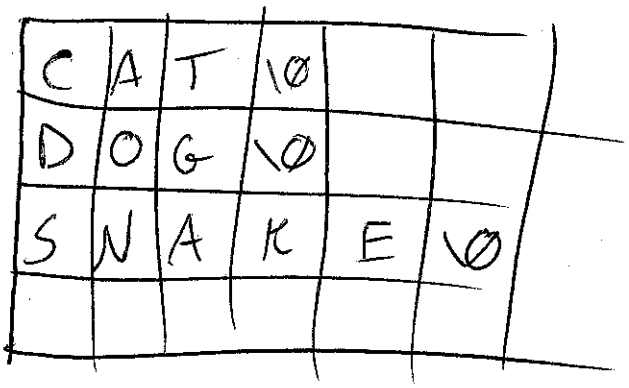
X = (int*) realloc(X, 10 * sizeof(int))

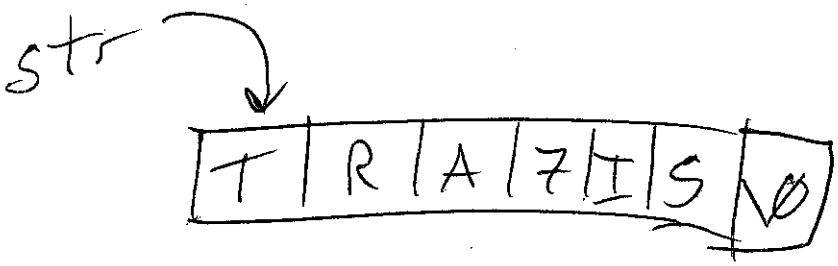


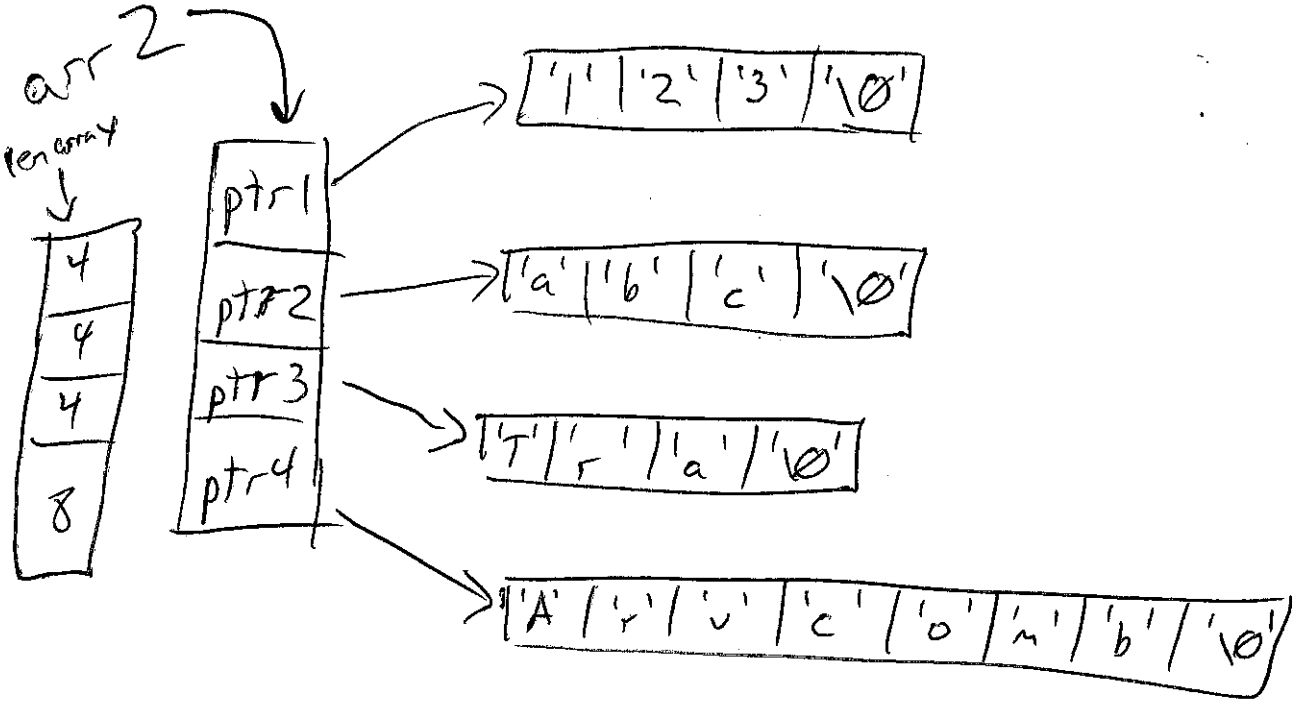


X = (int*) calloc (7, sizeof(int)); ← will fill with 8,6,7,3,5,1,0

```
int *tmp;
tmp = (int *) calloc (10, sizeof(int));
for(i=0; i<7; i++){
    tmp[i] = x[i];
} free(x); x = tmp;
```



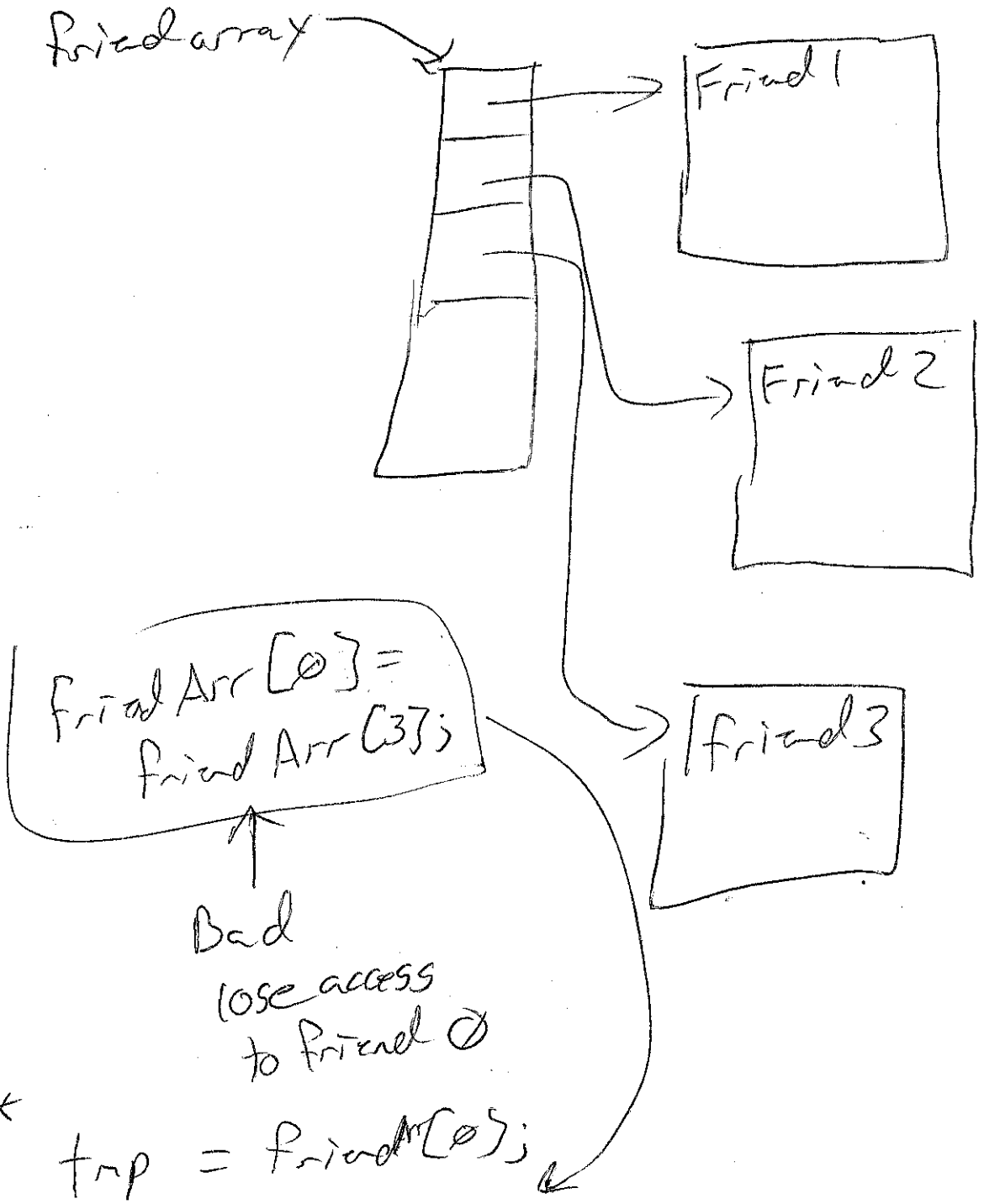




int array * int

char array * char

char array array * (* char)



~~friendArr[3] = tmp~~

friendArr[3] = trp;

Don't free trp