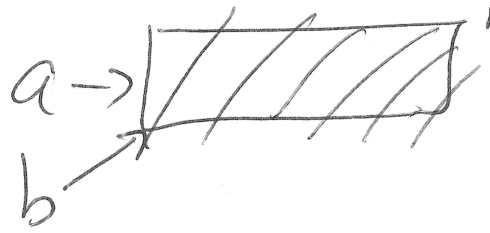
COP 3502 1/11/24

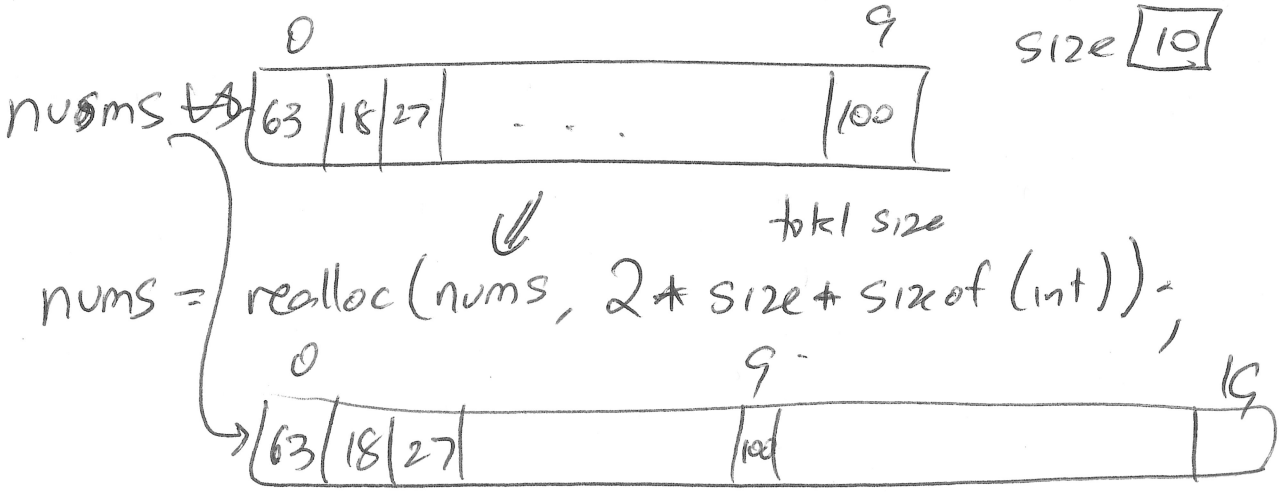
malloc
free
calloc
realloc

```
int* arr = calloc( 3*n , sizeof(int))
// allocates an array o with 3n slots each
// of which is an int AND initializes to
// all 0s.
```




```
free(b)
free(a); // CRASH PROGRAM
```

COMMON ERROR




```
free(nums)    same
nums = realloc(nums, 0)
```

array of struct (mystruct)

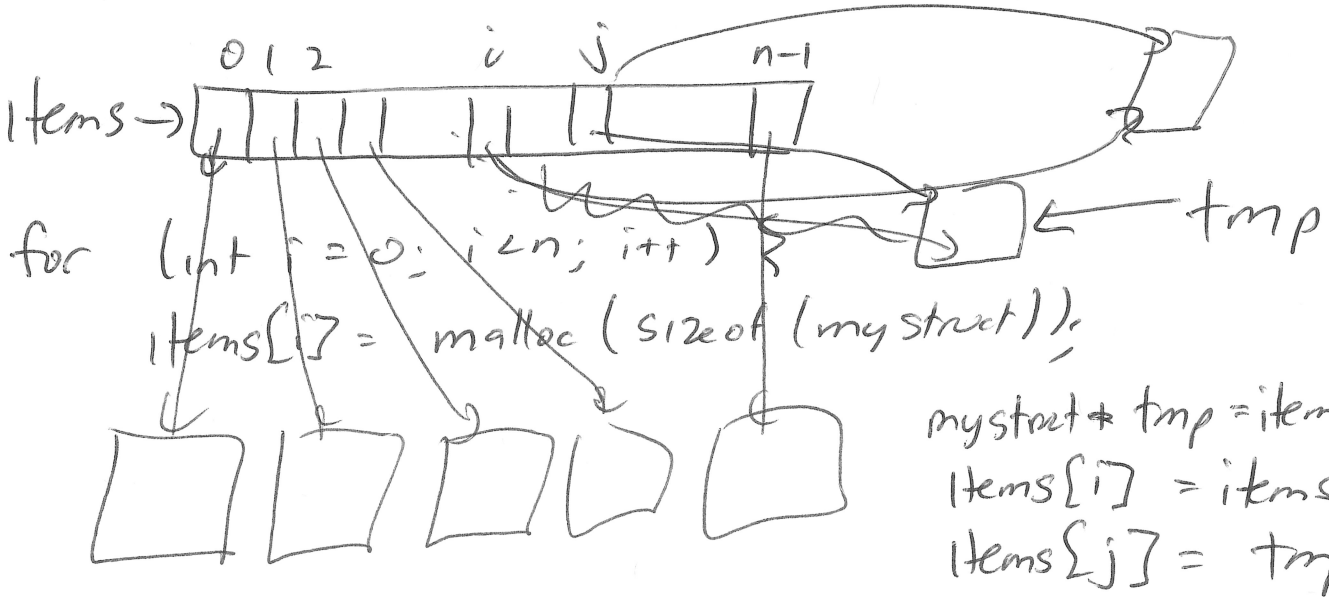
```
char name[30];
int num;
```

```
mystruct* items = calloc(n, sizeof(mystruct));
```


```
for (int i=0; i<n; i++)
    scanf("%s %d", items[i].name,
                   &items[i].num);
```

each malloc equal opposik free.

```
mystruct** items = calloc(n, sizeof(mystruct*));
```




```
for (int i=0; i<n; i++)
    items[i] = malloc(sizeof(mystruct));
```

```
mystruct* tmp = items[i]
items[i] = items[j];
items[j] = tmp;
```

equivalent

Struct.

ptrtostruct->

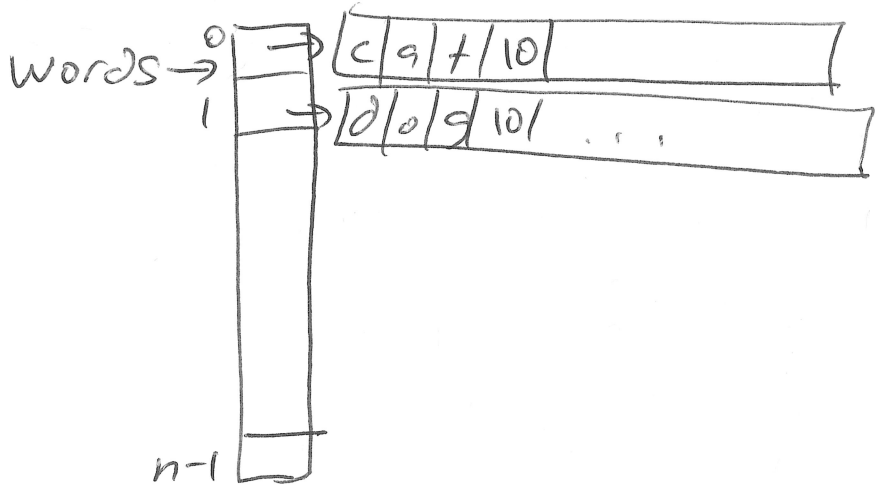
(*ptrtostruct).

} accessing fields

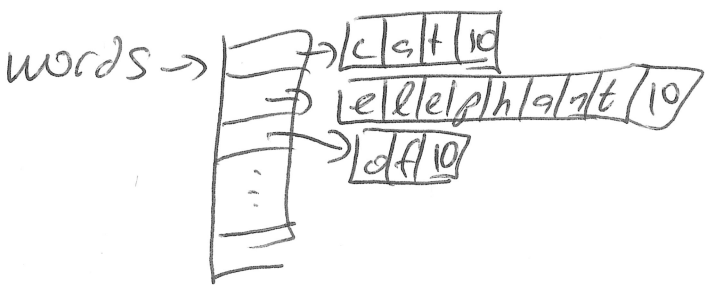
Storing an array of words

let's say I know all words are length 29 or less.

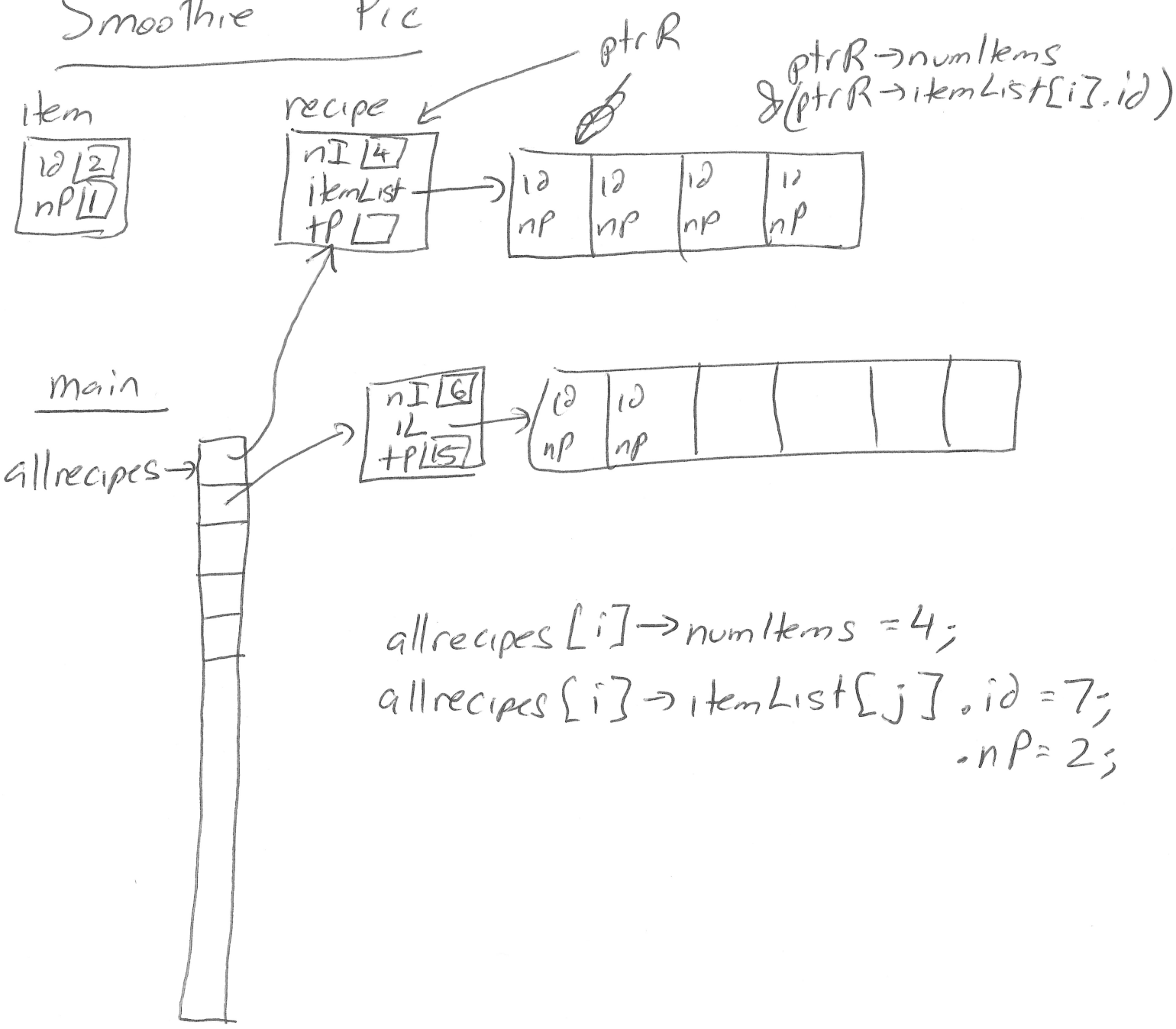
```
char** words = calloc (n, sizeof (char *));
for (int i=0; i< n; i++)
    words[i] = calloc (30, sizeof (char));
    scanf ("%s", words[i]);
}
```




```
char* tmp = malloc (100001 * sizeof (char));
char** words = calloc (n, sizeof(char*));
for (int i=0; i< n; i++) {
    scanf ("%s", tmp);
    words[i] = calloc ( strlen(tmp) +1, sizeof (char));
    strcpy (words[i], tmp);
}
```



Smoothie Pic



allrecipes[i]->numItems = 4;

allrecipes[i]->itemList[j].id = 7;

.nP = 2;

Sorted List Matching Problem

	length
a: 3, 6, 7, 22, 29, 40, 80	n
b: 2, 5, 6, 8, 9, 10, 15, 29, 32, 33, 60, 62, 64	m

```
int res = 0;
for (int i=0; i<n; i++) {
    int found=0;
    for (int j=0 ; j<m; j++)
        if (a[i] == b[j])
            found = 1;
    res += found;
}
```

$O(nm)$

```
int res=0;
for (int i=0; i<n; i++)
    res += binsearch(b, m, a[i]); // search array b of length m for a[i];
```

$O(n \cdot \lg m)$

---

a: 3, 6, 7, 22, 29, 40, 80

b: 2, 5, 6, 8, 9, 10, 15, 29, 32, 33, 60, 62, 64

```
int i=0, j=0;
while (i<n && j<m) {
    if (a[i] < b[j]) i++;
    else if (a[i] > b[j]) j++;
    else {
        printf("%d ", a[i]);
        i++;
        j++;
    }
}
```

O(m+n)

Sidebar: malloc too much space

$ptr = malloc(50 * sizeof(?) - 1)$

Should have been $ptr = malloc(50 * (sizeof(?) - 1))$