**Pointer comparisons**

```
char *s1 = "New Haven";
char *s2 = malloc(sizeof(char) * (strlen(s1) + 1));
char *s4 = malloc(sizeof(char) * (strlen(s1) + 1));
strcpy(s2, "New");
strcat(s2, " Haven");
char *s3 = "New Haven";
s1 = "West Haven";
```

checks whether ptrs point to same place
(same address)

```
if (s1 == s2)
{
    printf("Equal\n");
}

if (strcmp(s1, s2))
{
    // we will get here
}

if (s1 == s3)
{
    // will we get here?
}
```

YES

s1

New H ...

s2

W e s t ...

s3

s4

New H.

different places !
!=

New H...

```
track_list l;
l.size = 0;
l.capacity = 10;
l.starts = malloc(sizeof(int) * l->capacity);
l.ends = malloc(sizeof(int) * l->capacity);
```
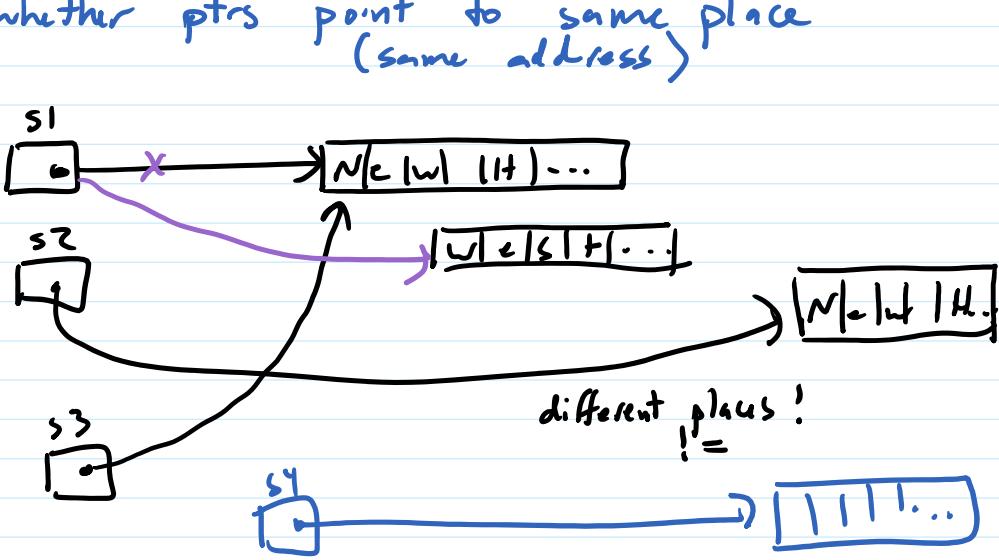
with opaque struct, don't know how big this needs to be!
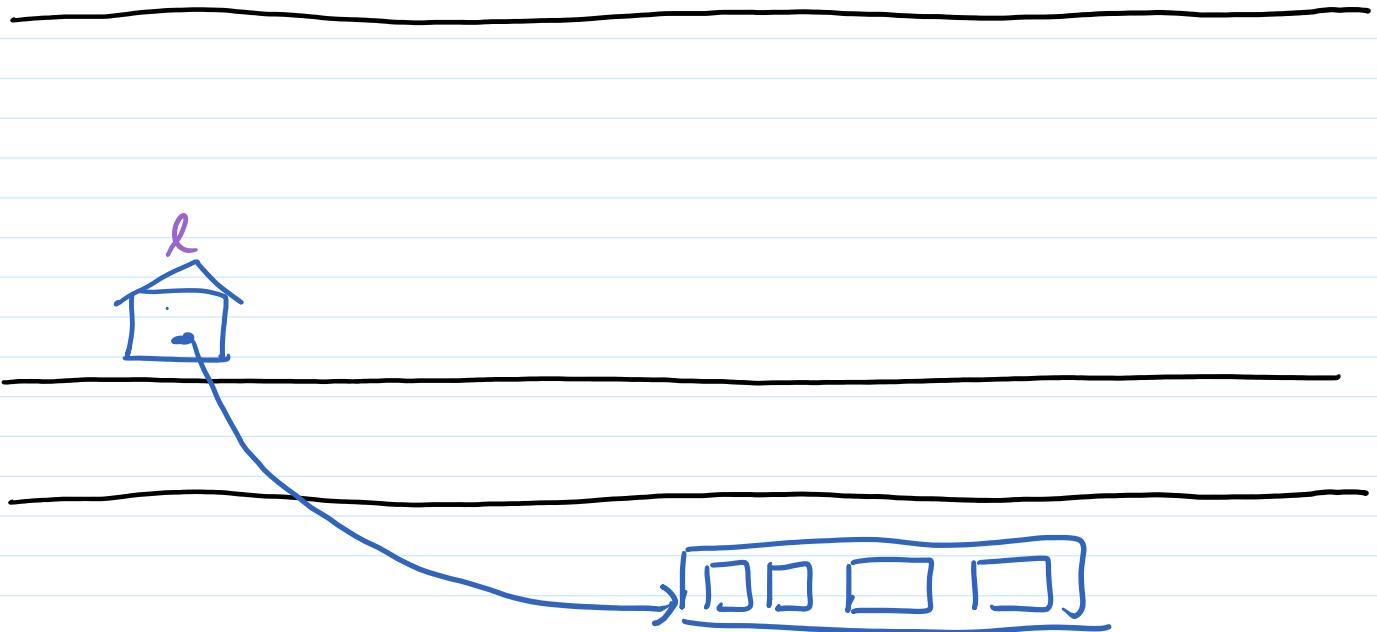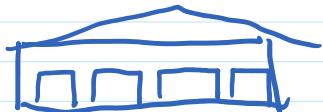
```
track_list *l;
l = malloc(sizeof(track_list));
l->size = 0;
...

track_list *l = track_list_create();
```

sizeof computed at compile time
so still don't know size

Abstract Data Type (ADT)
= specification of fxns available for type
   w/o spec of how those fxns work