Wrap up C:

```
s = realloc (len + 1);
```

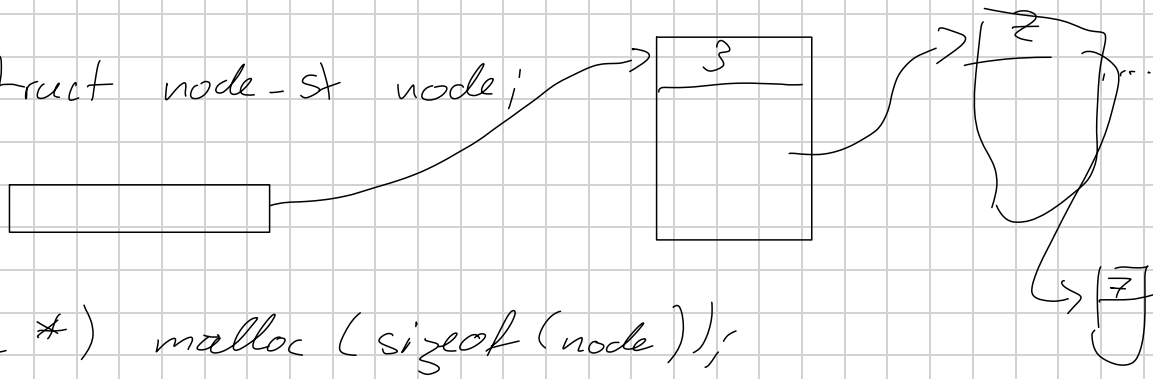
close (FILE *); ← realloc

structures

↳ an association of heterogeneous types

→ dereference & select

```
struct node_st
{
    int num;
    struct node_st *next;
};
struct node_st n;

n.num = 7;

typedef struct node_st node;

node *list;

list = (node *) malloc (sizeof (node));
(*list).num = 3;
(*(*(*list).next)).next).num = 7;
list->next->next->num = 7;
```



```
void printlist(node *l)
{
    for(;l;l=l->next){
        printf("%d\n",l->num);
    }
}
```

Odds & Ends

Enumerated types:

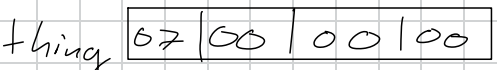
```
enum {red, green, blue} color;

typedef enum {false, true} boolean;
```

short-circuit evaluation:

```
thing *p;
if(p && *p)
```

unions:

↳ like a struct but all fields occupy the same space.

thing 


```
union {
    int num;
    char byte[sizeof(num)];
} thing;
thing.num = 7;
for(i=0; i<sizeof(int); i++) {
    printf("%d\n", thing.byte[i]);
```

comma:

exp1, exp2

value is exp2

```
for(i=0, j=2; ....
float pi = (3.14159);
```