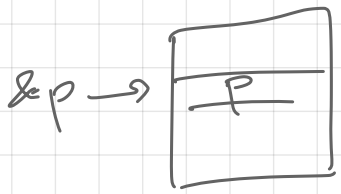
Week 1



Question 9.

```
int x;    // variable at address 1000 with initial value 0.
int *p;   // variable at address 2000 with initial value 0.
```

	statement	x value	p value	x address	p address
initial.	-	0	0	1000	2000
a.	p = &x;	0	1000	1000	2000
b.	x = 5;	5	1000	1000	2000
c.	*p = 3;	3	1000	1000	2000
d	x = (int) p;	1000	1000	1000	2000
e.	x = (int) &p;	2000	1000	1000	2000
f.	p = NULL;	2000	NULL	1000	2000
g.	*p = 1;		fails	because	p is NULL.

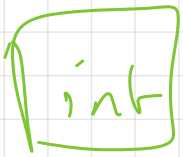
Question 6.

when to use * and/or malloc for structs?

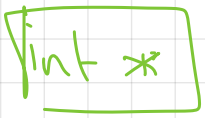
struct node a; use .
- declared on the stack
- don't need malloc.
- only use within a function.

struct node *b; use ->
- declared on the heap
- need malloc
- good for passing between functions

a pointer is a variable that points to the address of a variable.



var1;



var

struct

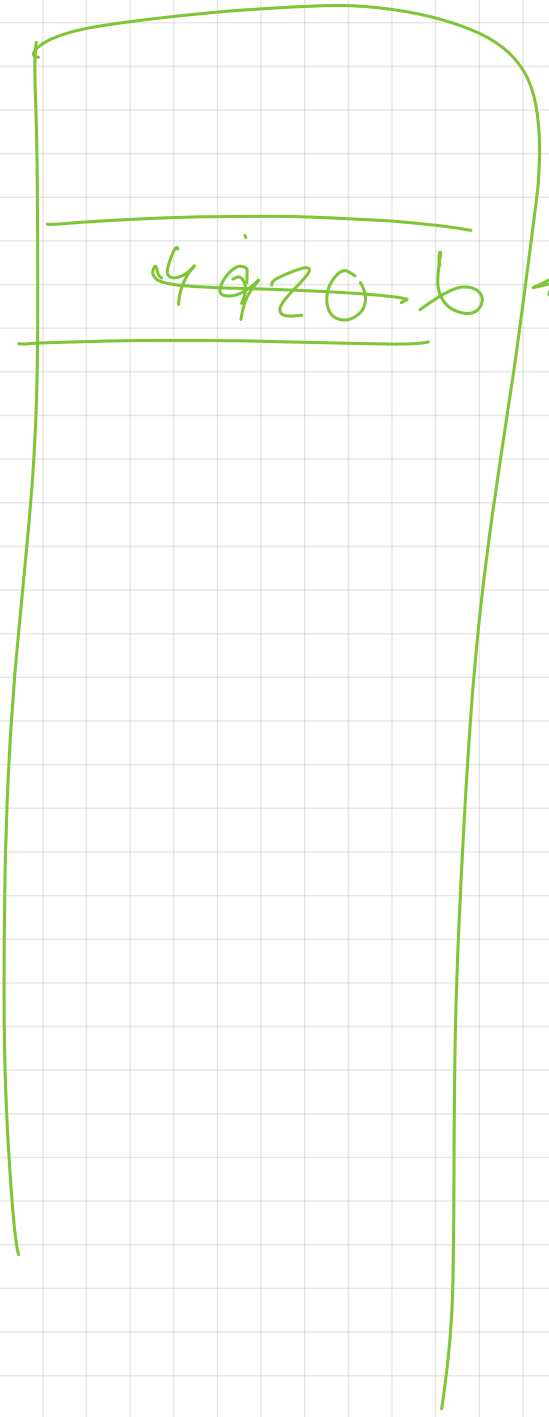
what does sizeof do?

sizeof gets the # of bytes (of variable eg "a" or type eg "int")

what does malloc do?

allocates memory of given size @ the address on LHS.

c = malloc (8) allocates 8 bytes @ c's address
- c: variable to allocate for
- 8: # of bytes



$f(5) = f(4) + f(3)$

$f(3)$ $f(2)$ $f(2)$ $f(1)$

r(0)
↓
r(1)
↓
r(2)
↓
3
4
⋮
10