Pointers

Missing C Features

For now, please avoid using the following two C syntactic constructs:

- & (address of)
- `void *`

Dereferencing A Pointer

```
int *ip = malloc(sizeof(int));

// To set the value pointed to by ip to 10, do either of the following
ip[0] = 10;
*ip = 10;

free(ip);
```

[ ]

int *

$(\text{int} *) \simeq (\text{int}[\,])$

[illegible]

int *p = 3;

int p2 = 3;

$\underbrace{\text{ip}}_{\text{int} *} \longrightarrow \boxed{}$

sizeof(ip)

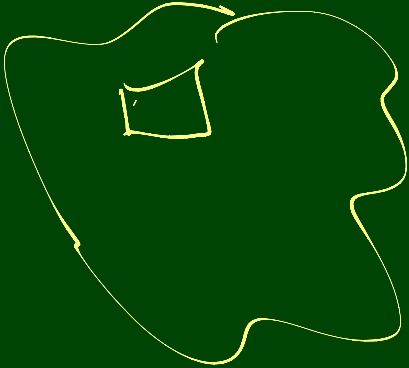
int * ≃ I am a pointer

follow the arrow → *p

Dereferencing A Pointer

```
int *ip = malloc(sizeof(int));

// To set the value pointed to by ip to 10, do either of the following
ip[0] = 10;
*ip = 10;

free(ip);
```

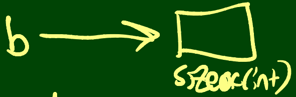
int i = 5;



Dereferencing A Pointer

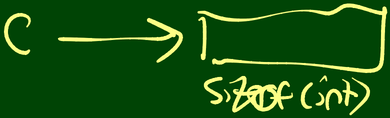
```c
int *ip = malloc(sizeof(int));

// To set the value pointed to by ip to 10, do either of the following
ip[0] = 10;
*ip = 10;

free(ip);
```

int *b = malloc(sizeof(int))



$\boxed{\text{int}}$ c[1];

c[0]



Dereferencing A Pointer

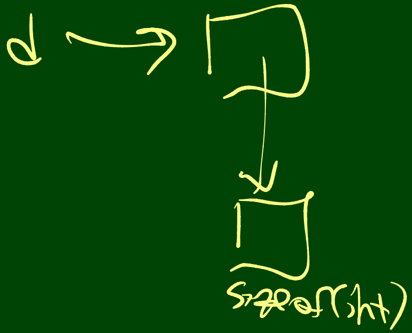
```c
int *ip = malloc(sizeof(int));

// To set the value pointed to by ip to 10, do either of the following
ip[0] = 10;
*ip = 10;

free(ip);
```

int *a = malloc(...);

int **d = malloc(sizeof(int *));

int **

(int*)*



C vs. Java, Again

Pixel in Java

```
public class Pixel {
   public int red;
   public int green;
   public int blue;

   public void zeroRed(Pixel p) {
      p.red = 0;
   }
}
```

pixel_t in C

```
typedef struct pixel {
   uint8_t red;
   uint8_t green;
   uint8_t blue;
} pixel_t;


void pixel_zero_red(pixel_t *p) {
   p->red = 0;
}
```