Pointers

How to declare a pointer:

```
int *ptr;
char* ptr;
float* ptr
```

- Declaring a pointer is not enough.
- It is important to initialize pointer before use.
- One way to initialize a pointer is to assign address of some variable.

how to initialize a pointer:

```
int x = 5
int* ptr;
ptr = &x;
```

- ***`ptr = &x`*** means assing address of variable `x` to pointer variable `ptr`.
- We can also write the whole thing in a single line:

```
int x = 5, *ptr = &x
```

Value of operator:

Value of operator/*indirection* operator/*dereference* operator that is used to access the value stored at the location pointed by the pointer.

```
int x = 5;
int* ptr;
ptr = &x;
printf("%d", *ptr); => 5
```

General rule:

if p points to `a[i]`, then:

```
p = p + j <=> &a[i + j]
```

```
p = p - j <=> &a[i - j]

a[i] = *(a + i)
```

Example:

The following code can be written in two different ways:

- ```
  for (p = a; p < &a[4]; p++) {

  }
  ```
- ```
  for (p = a; p < a + 4; p++) {

  }
  ```

Row vs Column major:

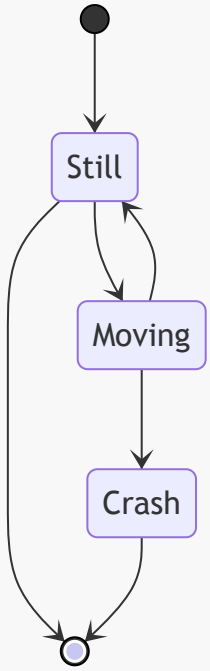
C stores multidementional arrays in row major order.
Using pointers we can transform these loops:

```
for (int i = 0; i < row; i++) {
    for (int j = 0; j < col; j++) {
        printf("%d ", a[i][j]);
    }
}
```

into a single loop:

```
for (p = &a[0][0]; p <= &a[row - 1][col - 1]; p++) {
    printf("%d ", *p);
}
```

State Diagram


Still
Moving
Crash