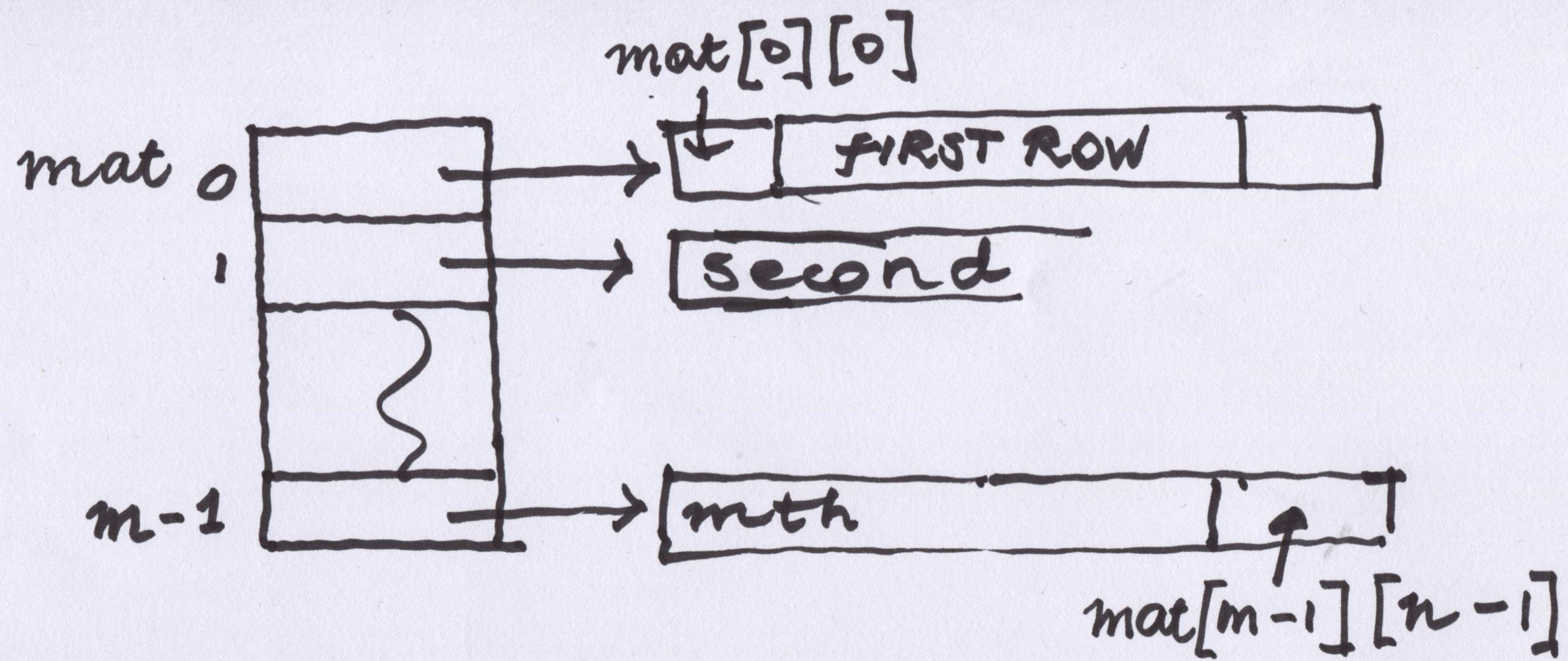




ARITHMETIC EXPRESSIONS

EG $1 - (2 - 3 * 4) / 5 + 6$

'Recursively' evaluate parenthesised

$$2 - 3 * 4$$

* trumps -
Evaluate $3 * 4$

$$2 - 12 = -10$$

$$1 - (-10) / 5 + 6$$

/ trumps -, +

$$1 - (-2) + 6$$

-, + at same precedence.

- is left most; $1 - (-2) = 3$

$$3 + 6$$

Result 9.

CONVERSIONS

double x = 1; printf("%f\n", x); 1.000000

int x = 1.23, y = -2.34; printf("x %d y %d\n", x, y);

char x = '7'; printf("x %c x %d x %d", x, x, x - '0');
x 7 x 55 x 7

Implicit types of constants

'\n' char

-45 int

1.23 double

'a' char

"hello" char *

2.00 double

Routine arguments converted

```
void print(double x)
{ printf("%f\n", x); }
int main()
{ print(23); }
```

a.out

23.000000

Casts

printf("%d\n", (int)(-2.34)); prints -2.

Pointers. Cast doesn't change value, changes meaning.

int num = 260; ~~int x = #~~ int *x = # prints

§17.7

char *y = (char *) x; printf("%d %d\n", *x, *y); 260 4

Some operators ~~have~~ ~~re~~ produce values.

```
int x=10; printf("x %d", x); printf("x++ %d\n", x++);  
          printf("x %d", x); printf("++x %d\n", ++x);  
          printf("x-- %d\n", x--);  
          " " printf("x-- %d\n", x--);  
          " " --x --x);
```

x	10	x++	10
x	11	++x	12
x	12	x--	12
x	11	--x	10

Precedence 1. highest. 9. Lowest

1. (LR) $\&a[i]$ $a[i]$ $x++$ $x--$
2. (RL) $++x$ $--x$ $*p$ $\&q$ $sizeof(type)$
3. (LR) $*x * y$ x / y $x \% y$
4. (LR) $x + y$ $x - y$
- 5 (LR) $x < y$ $x <= y$ $x >= y$ $x > y$
- 6 (LR) $x == y$ $x != y$
- 7 (LR) $x \&\& y$
- 8 (LR) $x \|\ y$
- 9 (RL) $x = y$ $x += y$ $x -= y$ etcetera

Exercises Fully parenthesise ~~to~~ according to precedence

$$1 - (2 - 3 * 4) / 5 + 6$$

$$(1 - ((2 - (3 * 4)) / 5)) + 6$$

while (*x++ != '\0') x++ level 1 * level 2 while ((*(x++)) != '\0')...

$$a = b = c == 0$$

$\uparrow$ $\uparrow$ $\uparrow$
 9 9 6

$$a = (b = (c == 0))$$

$$a = b == c = 0$$

$\uparrow$ $\uparrow$ $\uparrow$
 9 6 9

$a = ((b == c) = 0)$ ILLEGAL. Can't assign to expression

$$a = b = c == d \&\& e \parallel f \parallel g$$

9 9 6 7 8 8

$$a = (b = (((c == d) \&\& e) \parallel f) \parallel g))$$

$$a = (b = (((c == d) \&\& e) \parallel f) \parallel g))$$