Pointers and Arrays

Strong relationship between pointers and arrays in C

Declaration of an array with N elements:

[element type] [array name][N];

E.g., int a[10];
Access to the element $a(i-1)$

Notation: $a(i-1)$

Diagram:

- $a(0)$
- $a(1)$
- $a(2)$
\[ \text{int } \ast i \]
\[ a[i] \Leftarrow \ast (a+i) \]
\[ \& a[i] \Leftarrow a+i \]

\[ i = 13 \]
\[ a[i][i] = 8 \]

\[ pa = a \]
\[ pa++ \]

Legal

\[ a = pa \]
\[ a++ \]

Not legal
2D arrays are stored in row-major order. The column index (2nd) varies fastest as you step through memory.

```
int C[3][2];
```

```
<table>
<thead>
<tr>
<th>C0</th>
<th>C1</th>
<th>C2</th>
</tr>
</thead>
<tbody>
<tr>
<td>C0</td>
<td>C1</td>
<td>C2</td>
</tr>
<tr>
<td>C0</td>
<td>C1</td>
<td>C2</td>
</tr>
<tr>
<td>C0</td>
<td>C1</td>
<td>C2</td>
</tr>
</tbody>
</table>
```

normal pointer manipulation is legal

```c
int *p;
p = C;
p = p + 1;
*p = 5;
C[?][?][?] = 5
C(2)[0] = 5
```

This extends to more than 2 dimensions. First array index varies slowest, last index varies fastest.
arrays of arrays

typedef int int32_t; on 32-bit x86
typedef short int int32_t; on 64-bit x86

int32_t i;

for (i = 0; i < 24; i++) {
    ...
    i = -
}

typedef int32 complex [2];

1/ complex is user defined type,
   array of 2 ints x/
   
   complex D [3];
   int c [3] [2];
structs

struct cplx {
    int real;
    int imag;
}

A define cplx as: heterogeneous data structure w/ 2 elements x/

struct cplx var;
access fields via notation

Var. real = 0
Var. imag = 1

struct cplx E[3]; E array of 3 structs

E[2]. real = 7
E[1]. imag = 3
typedef struct complex {
    char label;
    int real;
    int imag;
} Complex;

Complex x;
Complex y;

x.label = 'E';
x.real = y.real;
x.imag = y.imag;
Add two complex numbers:

```c
void sumComplex ( Complex a, Complex b, Complex *s )
{
    (*s).real = a.real + b.real;
    (*s).imag = a.imag + b.imag;
    ... notation...
}
No. 1: \texttt{sum complex 2} (Complex a, Complex b, Complex x)

\begin{align*}
& \\
& s \rightarrow \text{label } = 111 \\
& s \rightarrow \text{real } = a \cdot \text{real } + b \cdot \text{real } \\
& s \rightarrow \text{imag } = a \cdot \text{imag } + b \cdot \text{imag } \\
& \text{return}
\end{align*}

3

No. 2: \texttt{sum complex 3} (Complex *a, Complex *b, Complex *x)

\begin{align*}
& \\
& s \rightarrow \text{label } = 111 \\
& s \rightarrow \text{real } = a \rightarrow \text{real } + b \rightarrow \text{real } \\
& s \rightarrow \text{imag } = a \rightarrow \text{imag } + b \rightarrow \text{imag } \\
& \text{return}
\end{align*}

3

Usage \texttt{complex x, y, z; sumcomplex (x, y, &z); sumcomplex3(&x, &y, &z);}