C Review
Pointers, Arrays, and I/O

C Advice

- Draw stuff out
  - Variables are boxes, pointers are arrows
- Give a type your variables!
- & returns a value whose type has one more star than the type of the variable
  - int quux;
  - int* baz = &quux;
- Execute the fundamental operations one at a time
  - variable lookup, pointer deference, etc

Tracing Pointers – Warm Up

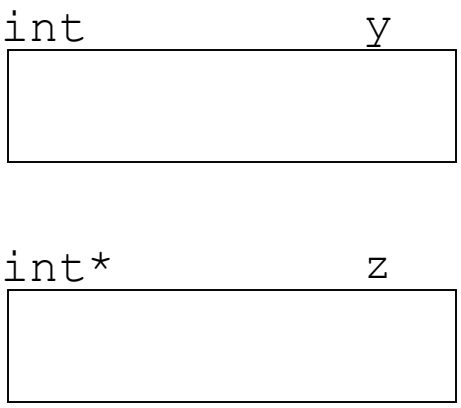
What will `y` contain?

```
int main(int argc, char** argv)
{
   int y, *z;
   y = 4;
   z = &y;
   y = *z + 9;
   return 0;
}
```

Tracing Pointers – Warm Up

What will `y` contain?

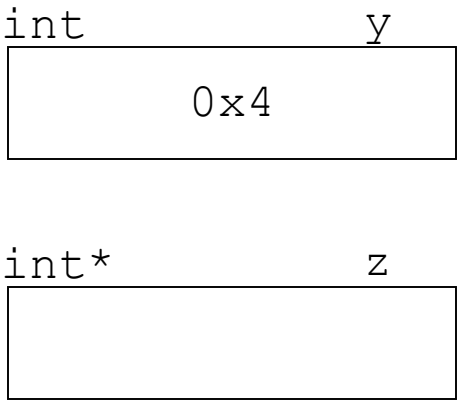
```
int main(int argc, char** argv)
{
   int y, *z;
   y = 4;
   z = &y;
   y = *z + 9;
   return 0;
}
```



Tracing Pointers – Warm Up

What will `y` contain?

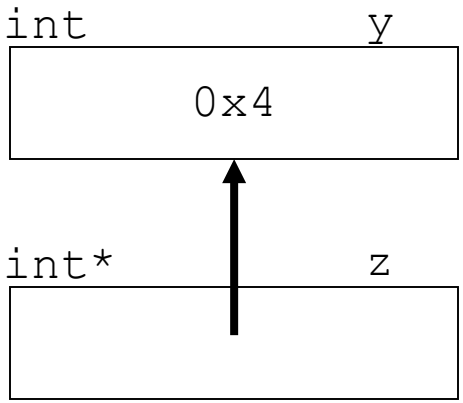
```
int main(int argc, char** argv)
{
   int y, *z;
   y = 4;
   z = &y;
   y = *z + 9;
   return 0;
}
```



Tracing Pointers – Warm Up

What will `y` contain?

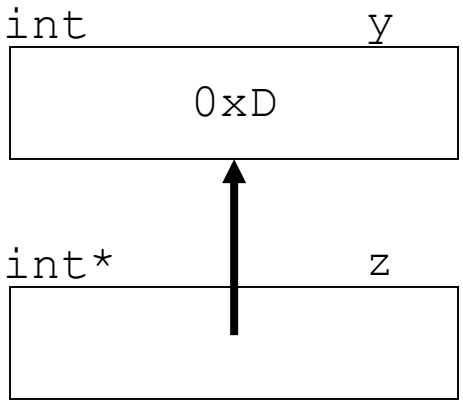
```
int main(int argc, char** argv)
{
   int y, *z;
   y = 4;
   z = &y;
   y = *z + 9;
   return 0;
}
```



Tracing Pointers – Warm Up

What will `y` contain?

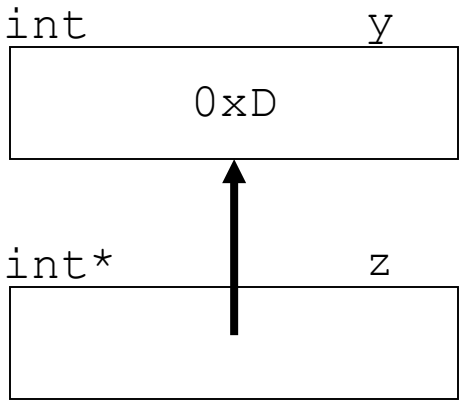
```
int main(int argc, char** argv)
{
   int y, *z;
   y = 4;
   z = &y;
   y = *z + 9;
   return 0;
}
```



Tracing Pointers – Warm Up

What will `y` contain?

```
int main(int argc, char** argv)
{
   int y, *z;
   y = 4;
   z = &y;
   y = *z + 9;
   return 0;
}
```



It contains 0xD. What is that in binary? In decimal?

Tracing Pointers – More Levels

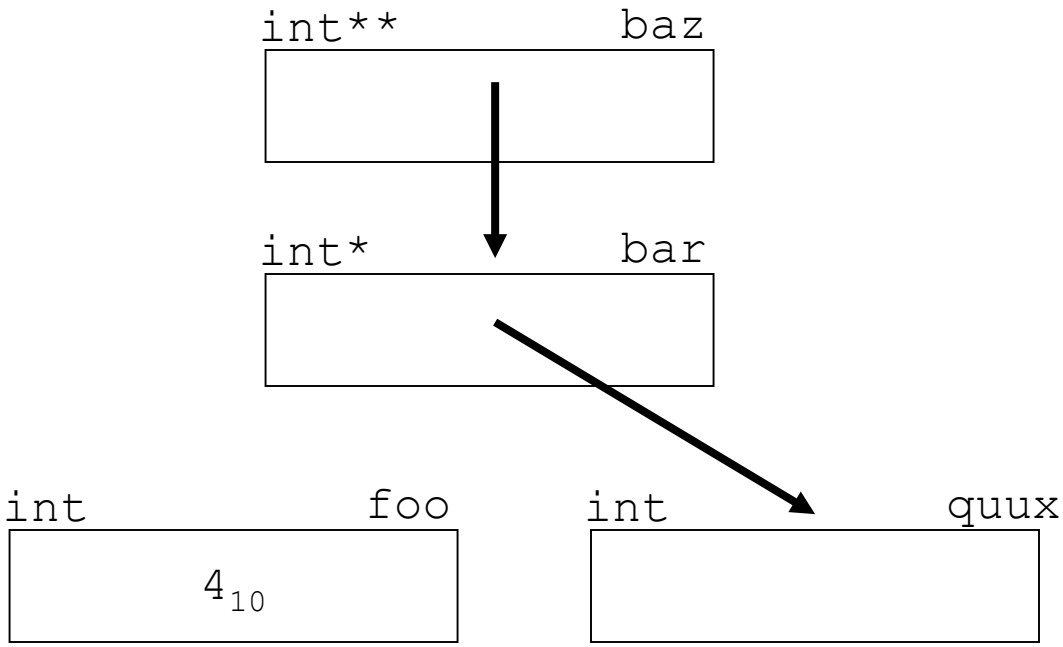
What is in foo and bar at the end of this program?

```
int main(int argc, char** argv)
{
  int foo, *bar, **baz, quux;
  bar = &quux;
  foo = 4;
  baz = &bar;
  **baz = 13;
  bar = &foo;
  **baz = 9;
  return 0;
}
```

Tracing Pointers – More Levels

What is in `foo` and `quux` at the end of this program?

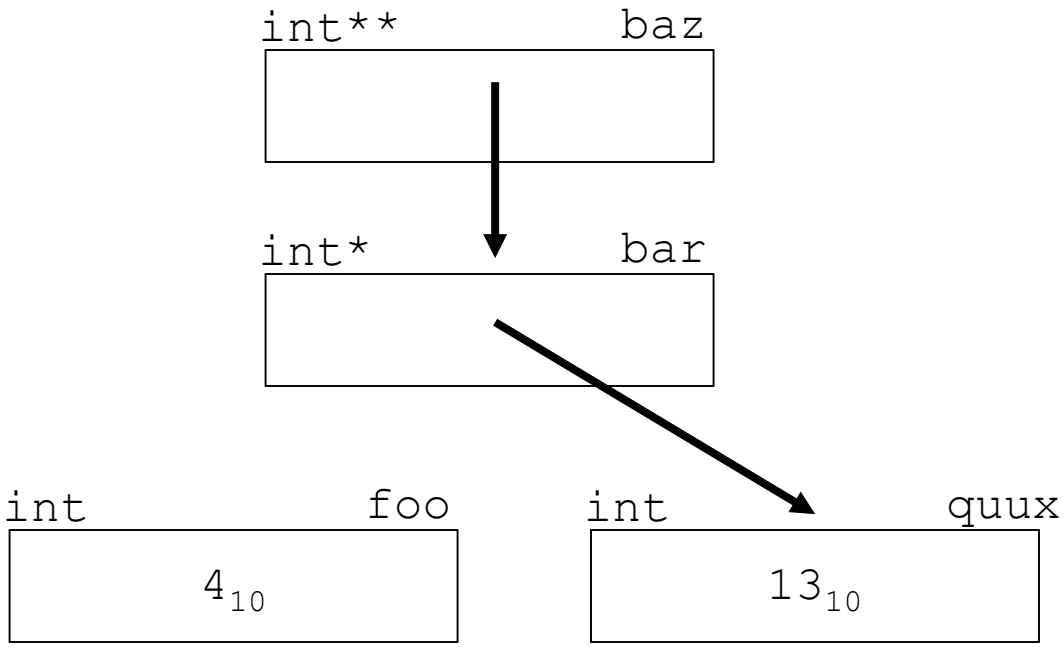
```
int main(int argc, char** argv)
{
  int foo, *bar, **baz, quux;
  bar = &quux;
  foo = 4;
  baz = &bar;
  **baz = 13;
  bar = &foo;
  **baz = 9;
   return 0;
}
```



Tracing Pointers – More Levels

What is in `foo` and `quux` at the end of this program?

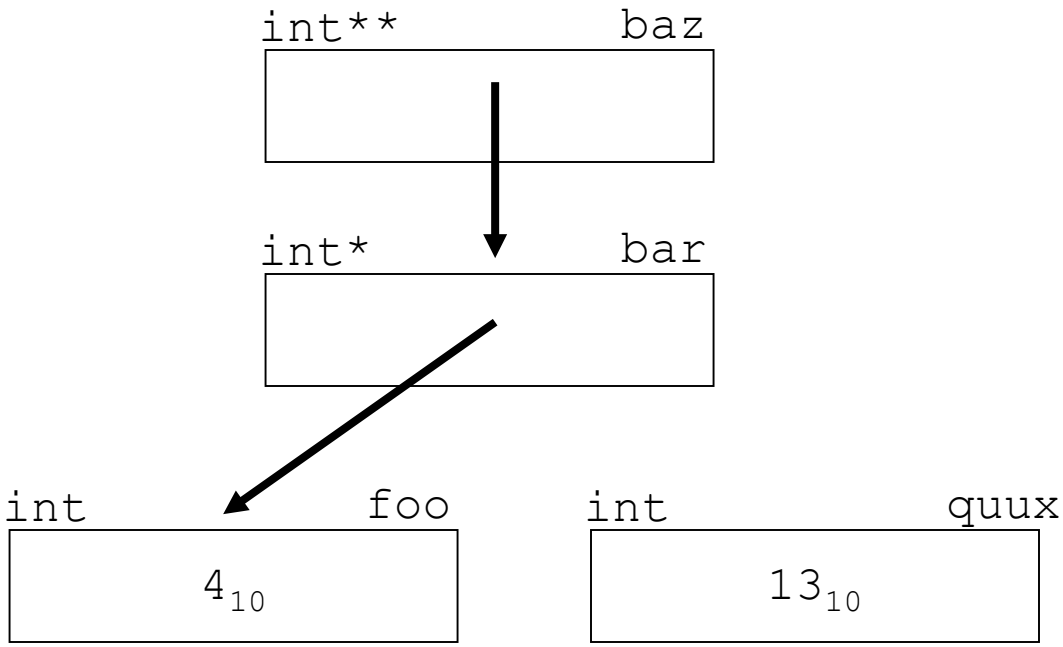
```
int main(int argc, char** argv)
{
  int foo, *bar, **baz, quux;
  bar = &quux;
  foo = 4;
  baz = &bar;
  **baz = 13;
  bar = &foo;
  **baz = 9;
   return 0;
}
```



Tracing Pointers – More Levels

What is in `foo` and `quux` at the end of this program?

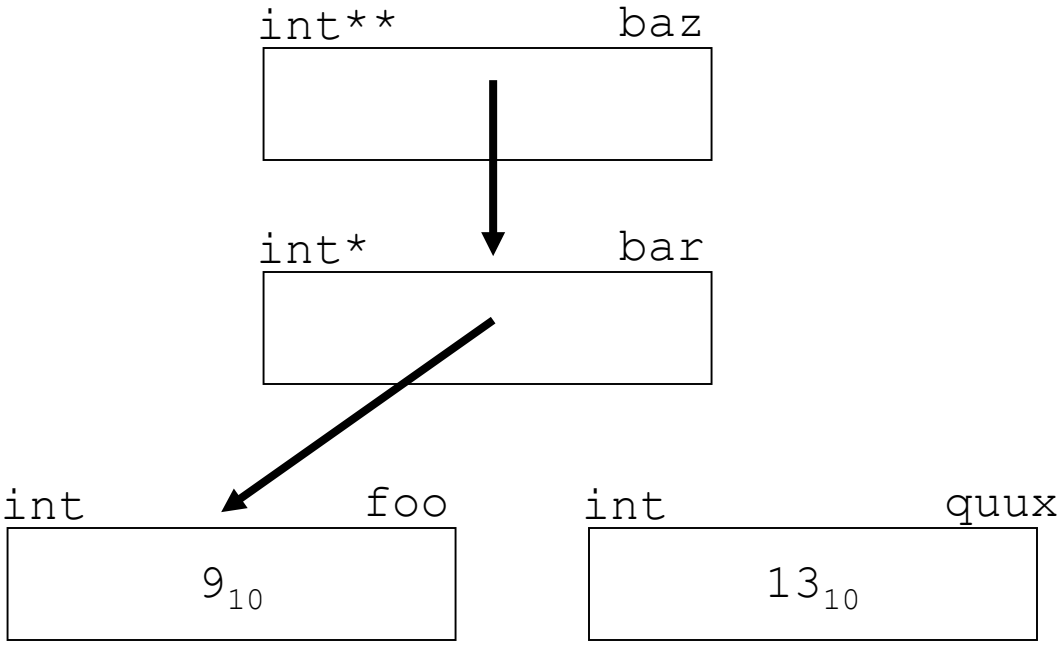
```
int main(int argc, char** argv)
{
  int foo, *bar, **baz, quux;
  bar = &quux;
  foo = 4;
  baz = &bar;
  **baz = 13;
  bar = &foo;
  **baz = 9;
   return 0;
}
```



Tracing Pointers – More Levels

What is in `foo` and `quux` at the end of this program?

```
int main(int argc, char** argv)
{
  int foo, *bar, **baz, quux;
  bar = &quux;
  foo = 4;
  baz = &bar;
  **baz = 15;
  bar = &foo;
  **baz = 9;
   return 0;
}
```



What's wrong with this program?

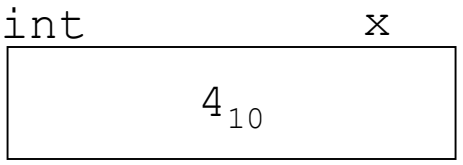
```
int modifyCount(int x)
{
  x = x - 1;
}

int main(int argc, char** argv)
{
  int x = 4;
  /* want to change x */
  modifyCount(x);
  return 0;
}
```

What's wrong with this program?

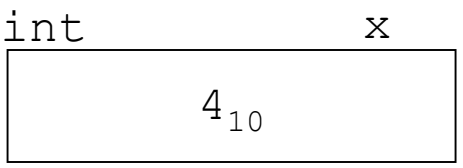
```
int modifyCount(int x)
{
  x = x - 1;
}

int main(int argc, char** argv)
{
  int x = 4;
  /* want to change x */
  modifyCount(x);
  return 0;
}
```

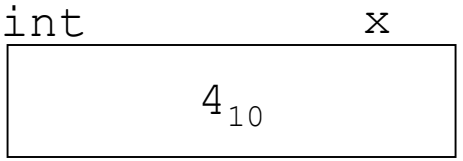


What's wrong with this program?

```
int modifyCount(int x)
{
  x = x - 1;
}
```

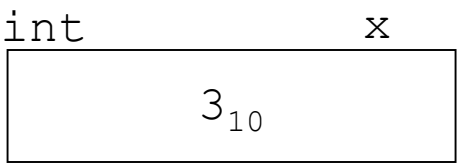


```
int main(int argc, char** argv)
{
  int x = 4;
  /* want to change x */
  modifyCount(x);
  return 0;
}
```

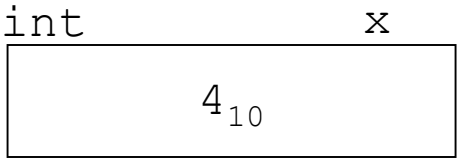


What's wrong with this program?

```
int modifyCount(int x)
{
  x = x - 1;
}
```



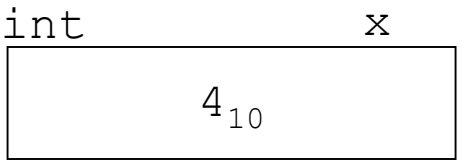
```
int main(int argc, char** argv)
{
  int x = 4;
  /* want to change x */
  modifyCount(x);
  return 0;
}
```



What's wrong with this program?

```
int modifyCount(int x)
{
  x = x - 1;
}
```

```
int main(int argc, char** argv)
{
  int x = 4;
  /* want to change x */
  modifyCount(x);
  return 0;
}
```



We never changed `x`! How do we fix this?

Use Pointers!

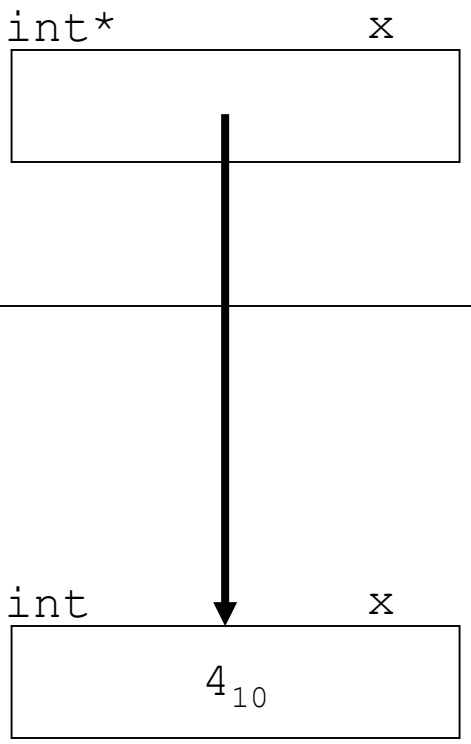
```
int modifyCount(int* x)
{
  *x = *x - 1;
}

int main(int argc, char** argv)
{
  int x = 4;
  /* want to change x */
  modifyCount(&x);
  return 0;
}
```

What's wrong with this program?

```
int modifyCount(int* x)
{
  *x = *x - 1;
}
```

```
int main(int argc, char** argv)
{
  int x = 4;
  /* want to change x */
  modifyCount(&x);
  return 0;
}
```



Pointers and ++/--

```c
#include <stdio.h>

int main(void)
{
  int age = 5, i = 0, j = 0;
  int* p = &age;
  i = (*p)++;                    // post associativity  is left to right , by the way higher than pre
  printf("\n%d", age);
  printf("\n%d", i);

  age = 55;
  j = ++(*p);                    // pre associativity  is right to left
  printf("\n%d", age);
  printf("\n%d", j);
  return 0;
}

```

input

```
6
5
56
56

...Program finished with exit code 0
Press ENTER to exit console.
```

Pointers and `[]`

- `x[i]` can always be rewritten as `*(x+i)` and vice versa

- Array types can often be converted into their corresponding pointer counterparts
  - `int foo[]` is equivalent to `int* foo`
  - `int* bar[]` is equivalent to `int** bar`
  - You can at most change one set of `[]` safely
    - Changing more requires knowing how the array looks in memory

printf, scanf, and their cousins

- printf (and its cousins) are special functions that do not have a fixed argument list
  - for each format specifier (i.e. %d), an additional argument needs to be supplied

- Examples:
  - `printf("%d", 4);`
  - `printf("%s%d%c", "CS", 0x3D, 'c');`

printf, scanf, and their cousins

- Unlike printf, with scanf for each format specifier (i.e. %d), an additional argument needs to be supplied that has type pointer

- Examples:
  - `int z; scanf(“%d”, &z);`
  - `char foo[5]; int d;`
    `scanf(“%s %d”, foo, &d);`