CS1010

Evan Tay

evantay@comp.nus.edu.sg

https://github.com/DigiPie/cs1010_tut_c09

Today’s plan

- Recap
- Problem Sets 13, 14, 15
- Consultation

evantay@comp.nus.edu.sg

RECAP

Pointers and Arrays

Pointer Fact #1: Short-cut

```
long *x = &y;
```

Equivalent to:

```
long *x;
x = &y;
```

Not equivalent to:

```
long *x;
*x = &y;
```


evantay@comp.nus.edu.sg

Pointer Fact #2: [] in different context

1. long a[10];
   - *declare an array of 10 elements*
2. a[10] = 1;
   - *set the 11th element of the array to 1*
3. foo(a[10]);
   - *pass the 11th array element into foo*
4. void foo(long a[10]) { .. }
   - *foo accepts an array of long as argument*

evantay@comp.nus.edu.sg

Pointer Fact #3: Array decay

1. long *a = cs1010_read_long_array(10); // OKAY
   - cs1010_read_long_array() *returns a pointer to an array, and not an array itself.*
   - *can still use* a[i] *which is equivalent to* *(a + i)
2. long *a = {1, 2, 3}; // NOT OKAY
   - **a is a pointer, no space is allocated for 3 long values*

https://stackoverflow.com/questions/1461432/what-is-array-decaying

evantay@comp.nus.edu.sg

PROBLEM SETS

13, 14, 15

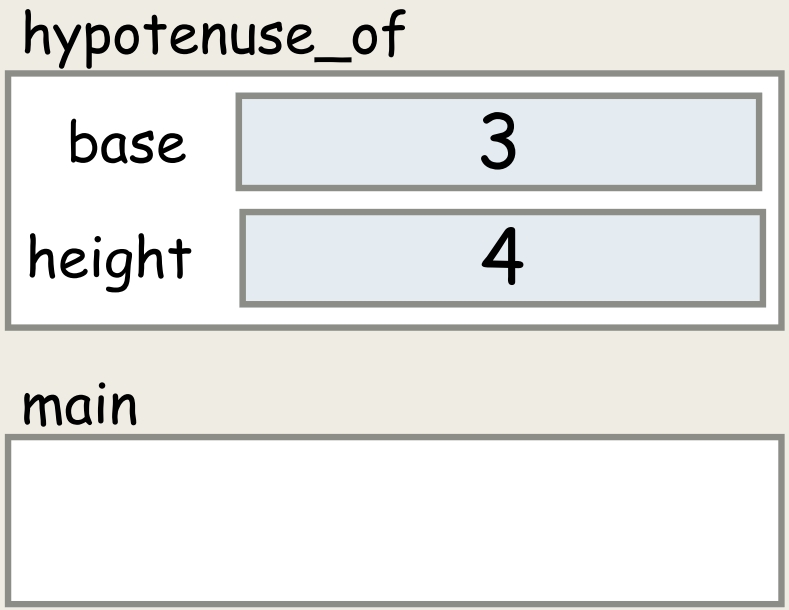
Problem Set 13.1

```
#include <math.h>
long square(long x) {
  return x*x;
}

double hypotenuse_of(long base, long height) {
  return sqrt(square(base) + square(height));
}

int main() {
  hypotenuse_of(3, 4);
}
```


evantay@comp.nus.edu.sg

Problem Set 13.1

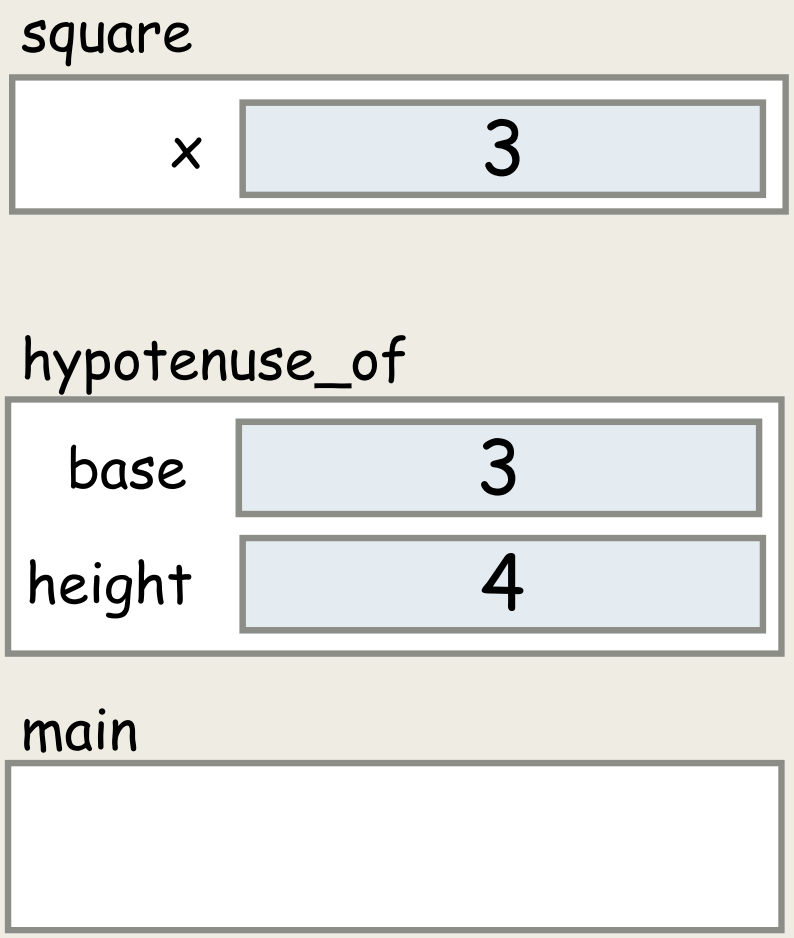
```
#include <math.h>
long square(long x) {
  return x*x;
}

double hypotenuse_of(long base, long height) {
  return sqrt(square(base) + square(height));
}

int main() {
  hypotenuse_of(3, 4);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.1

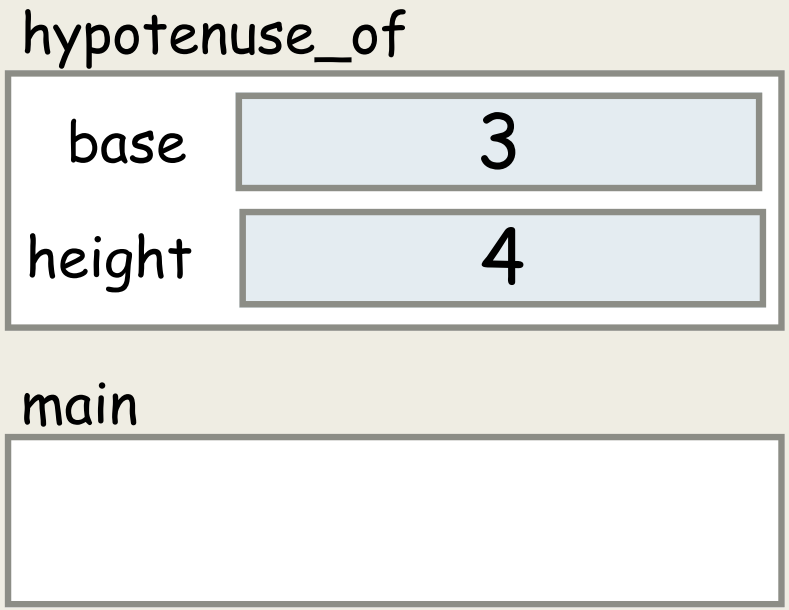
```
#include <math.h>
long square(long x) {
  return x*x;
}

double hypotenuse_of(long base, long height) {
  return sqrt(square(base) + square(height));
}

int main() {
  hypotenuse_of(3, 4);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.1

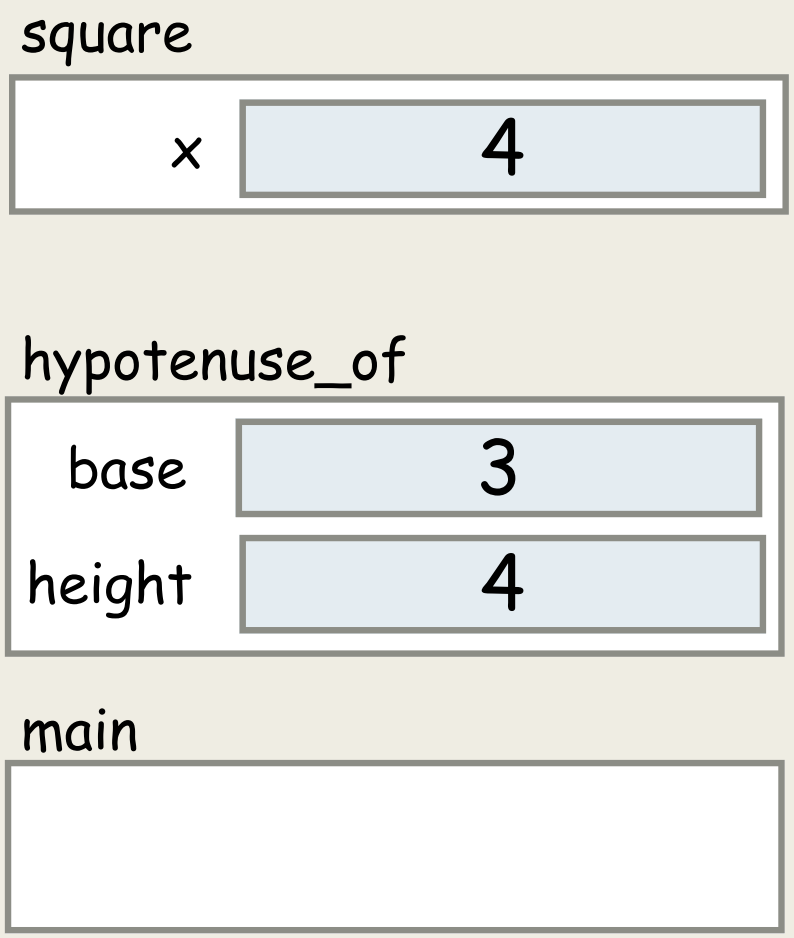
```
#include <math.h>
long square(long x) {
  return x*x;
}

double hypotenuse_of(long base, long height) {
  return sqrt(square(base) + square(height));
}

int main() {
  hypotenuse_of(3, 4);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.1

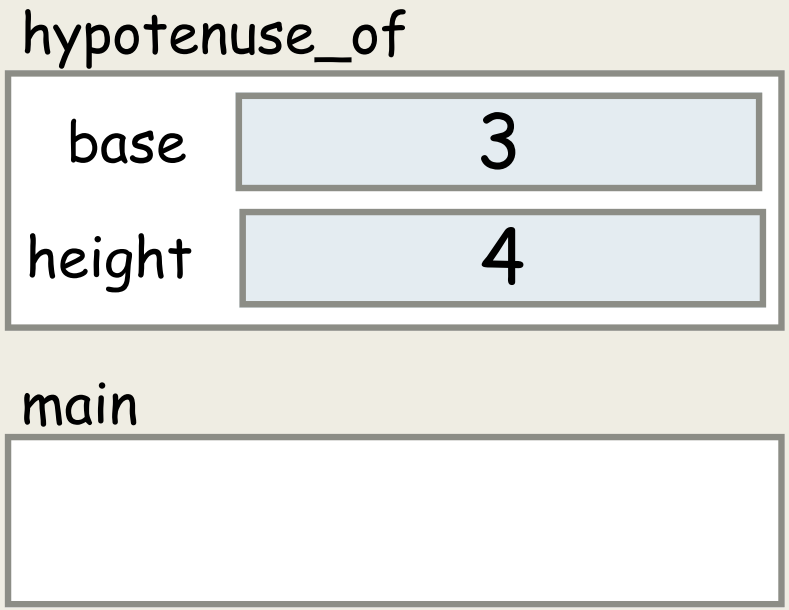
```
#include <math.h>
long square(long x) {
  return x*x;
}

double hypotenuse_of(long base, long height) {
  return sqrt(square(base) + square(height));
}

int main() {
  hypotenuse_of(3, 4);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.1

```
#include <math.h>
long square(long x) {
  return x*x;
}

double hypotenuse_of(long base, long height) {
  return sqrt(square(base) + square(height));
}

int main() {
  hypotenuse_of(3, 4);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.1

```
#include <math.h>
long square(long x) {
  return x*x;
}

double hypotenuse_of(long base, long height) {
  return sqrt(square(base) + square(height));
}

int main() {
  hypotenuse_of(3, 4);
}
```

sqrt

x	25

hypotenuse_of

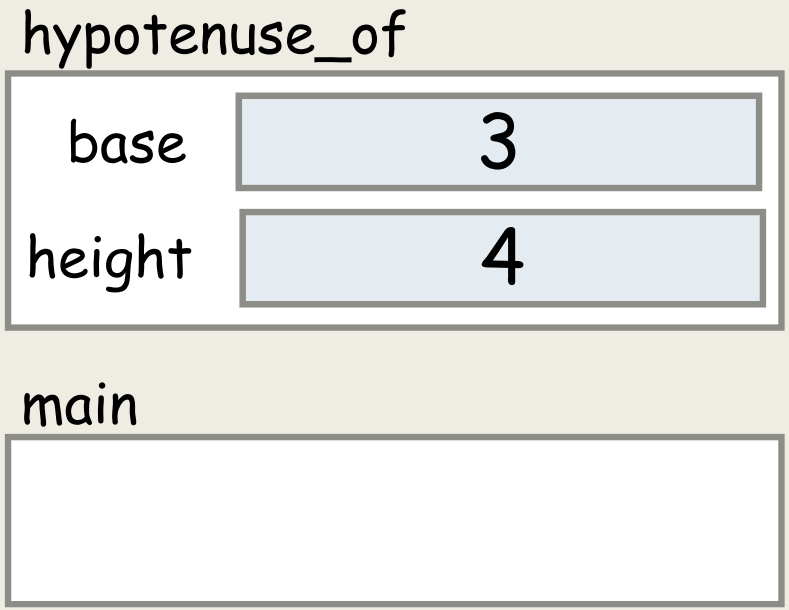
base	3
height	4

main

evantay@comp.nus.edu.sg

Problem Set 13.1

```
#include <math.h>
long square(long x) {
  return x*x;
}

double hypotenuse_of(long base, long height) {
  return sqrt(square(base) + square(height));
}

int main() {
  hypotenuse_of(3, 4);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.1

```
#include <math.h>
long square(long x) {
  return x*x;
}

double hypotenuse_of(long base, long height) {
  return sqrt(square(base) + square(height));
}

int main() {
  hypotenuse_of(3, 4);
}
```

main

evantay@comp.nus.edu.sg

PROBLEM SETS

13, 14, 15

Problem Set 13.2

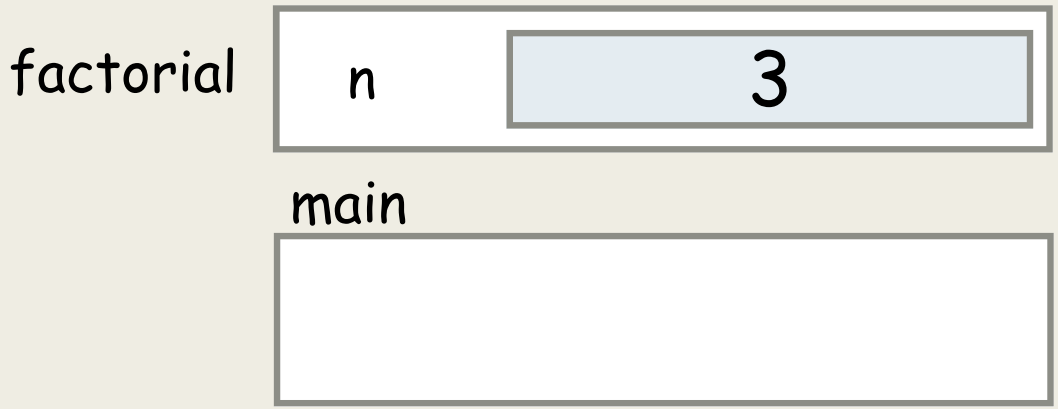
```
#include "cs1010.h"

long factorial(long n) {
  if (n == 0) {
    return 1;
  }

  return factorial(n-1) * n;
}


int main() {

  factorial(3);
}
```

main

evantay@comp.nus.edu.sg

Problem Set 13.2

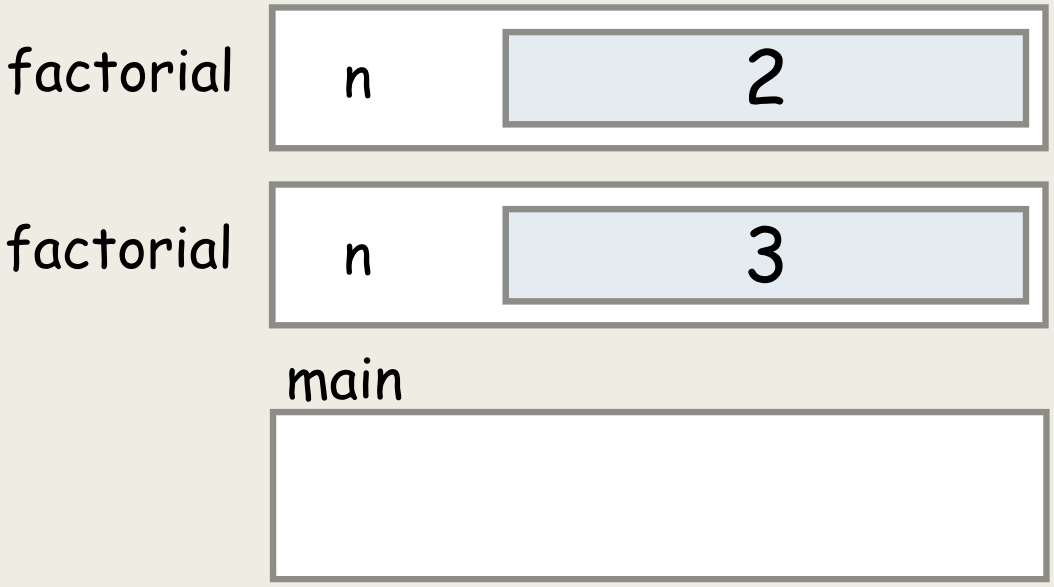
```
#include "cs1010.h"

long factorial(long n) {
  if (n == 0) {
    return 1;
  }

  return factorial(n-1) * n;
}


int main() {

  factorial(3);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.2

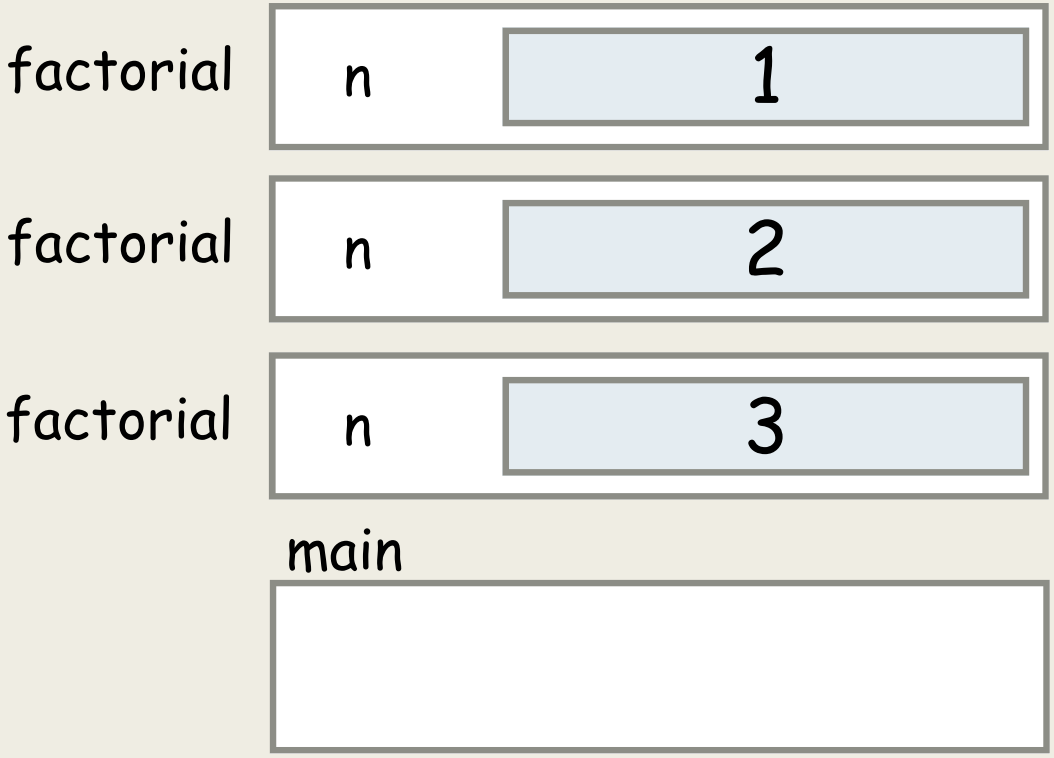
```
#include "cs1010.h"

long factorial(long n) {
  if (n == 0) {
    return 1;
  }

  return factorial(n-1) * n;
}


int main() {

  factorial(3);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.2

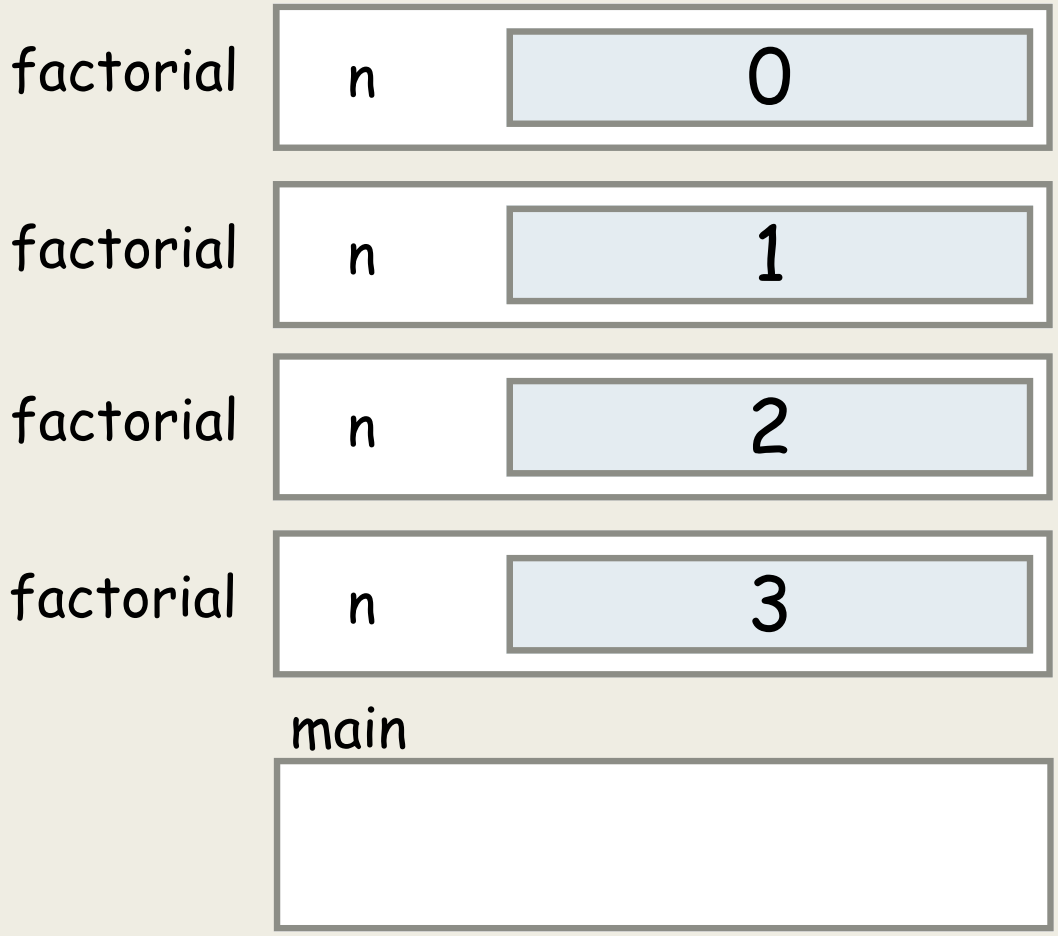
```
#include "cs1010.h"
long factorial(long n) {
  if (n == 0) {
    return 1;
  }
  return factorial(n-1) * n;
}

int main() {
  factorial(3);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.2

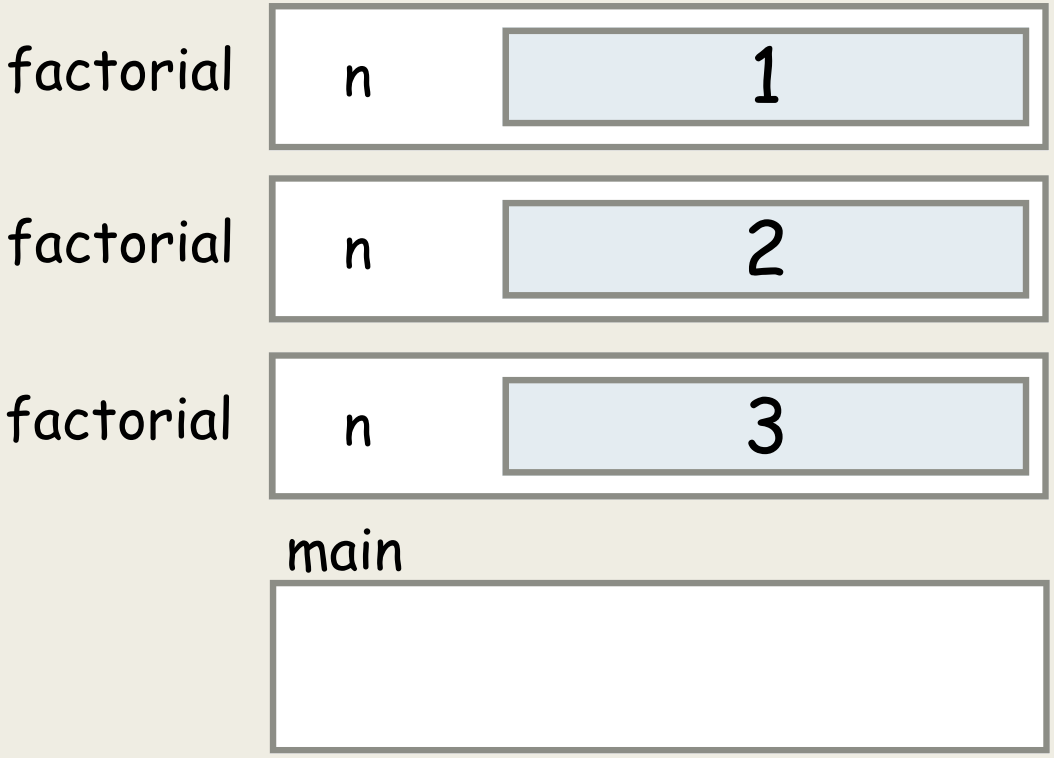
```
#include "cs1010.h"
long factorial(long n) {
  if (n == 0) {
    return 1;
  }
  return factorial(n-1) * n;
}

int main() {
  factorial(3);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.2

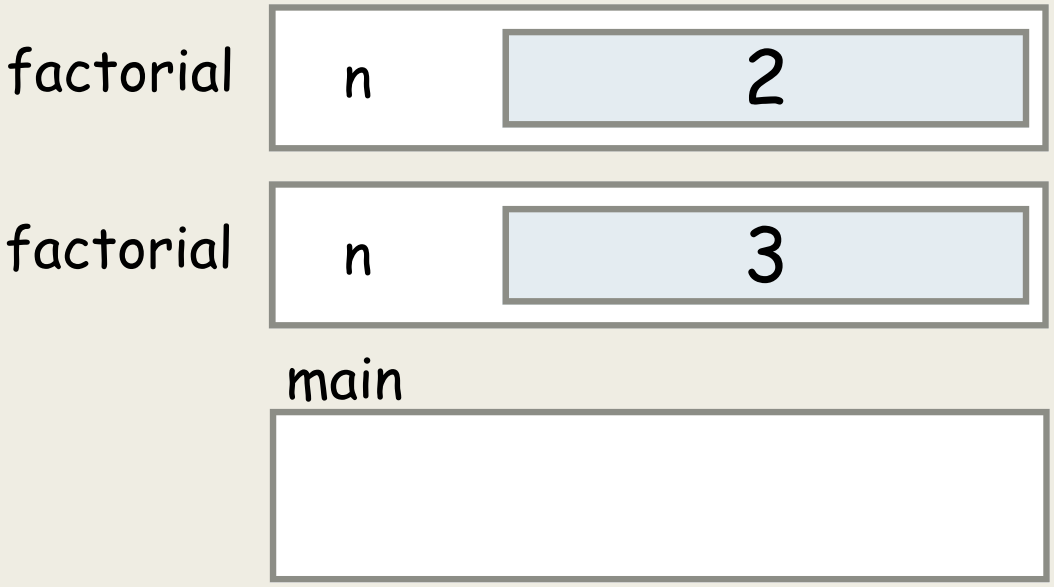
```
#include "cs1010.h"
long factorial(long n) {
  if (n == 0) {
    return 1;
  }
  return factorial(n-1) * n;
}

int main() {
  factorial(3);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.2

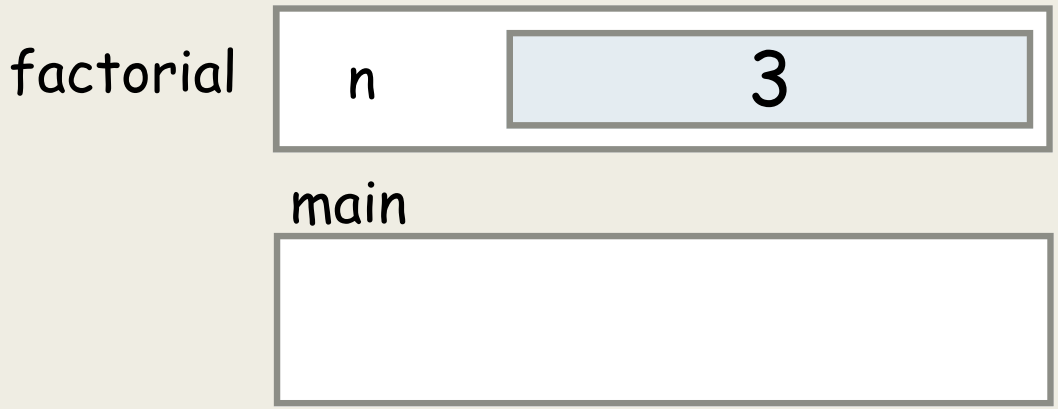
```
#include "cs1010.h"

long factorial(long n) {
  if (n == 0) {
    return 1;
  }

  return factorial(n-1) * n;
}


int main() {

  factorial(3);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.2

```
#include "cs1010.h"

long factorial(long n) {
  if (n == 0) {
    return 1;
  }

  return factorial(n-1) * n;
}


int main() {

  factorial(3);
}
```



evantay@comp.nus.edu.sg

Problem Set 13.2

```
#include "cs1010.h"

long factorial(long n) {
  if (n == 0) {
    return 1;
  }

  return factorial(n-1) * n;
}


int main() {

  factorial(3);
}
```

main

evantay@comp.nus.edu.sg

PROBLEM SETS

13, 14, 15

Problem Set 13.3

```
#include "cs1010.h“

void incr(long x) {
  x += 1;
}


int main() {
  long x = 10;
  incr(x);
  incr(x);
  cs1010_print_long(x); // What will this print?
}
```


evantay@comp.nus.edu.sg

Problem Set 13.3

```
#include "cs1010.h“

void incr(long x) {
  x += 1;
}


int main() {
  long x = 10;
  incr(x);
  incr(x);
  cs1010_print_long(x); // What will this print? 10
}
```


evantay@comp.nus.edu.sg

PROBLEM SETS

13, 14, 15

Problem Set 14.1

```
long *ptr1;
long *ptr2;
long x;
long y;

ptr1 = &x;
ptr2 = &y;

*ptr1 = 123;
*ptr2 = -1;
```

```
cs1010_println_long(x); // Prints:
cs1010_println_long(y); // Prints:
cs1010_println_long(*ptr1); // Prints:
cs1010_println_long(*ptr2);  // Prints:
```

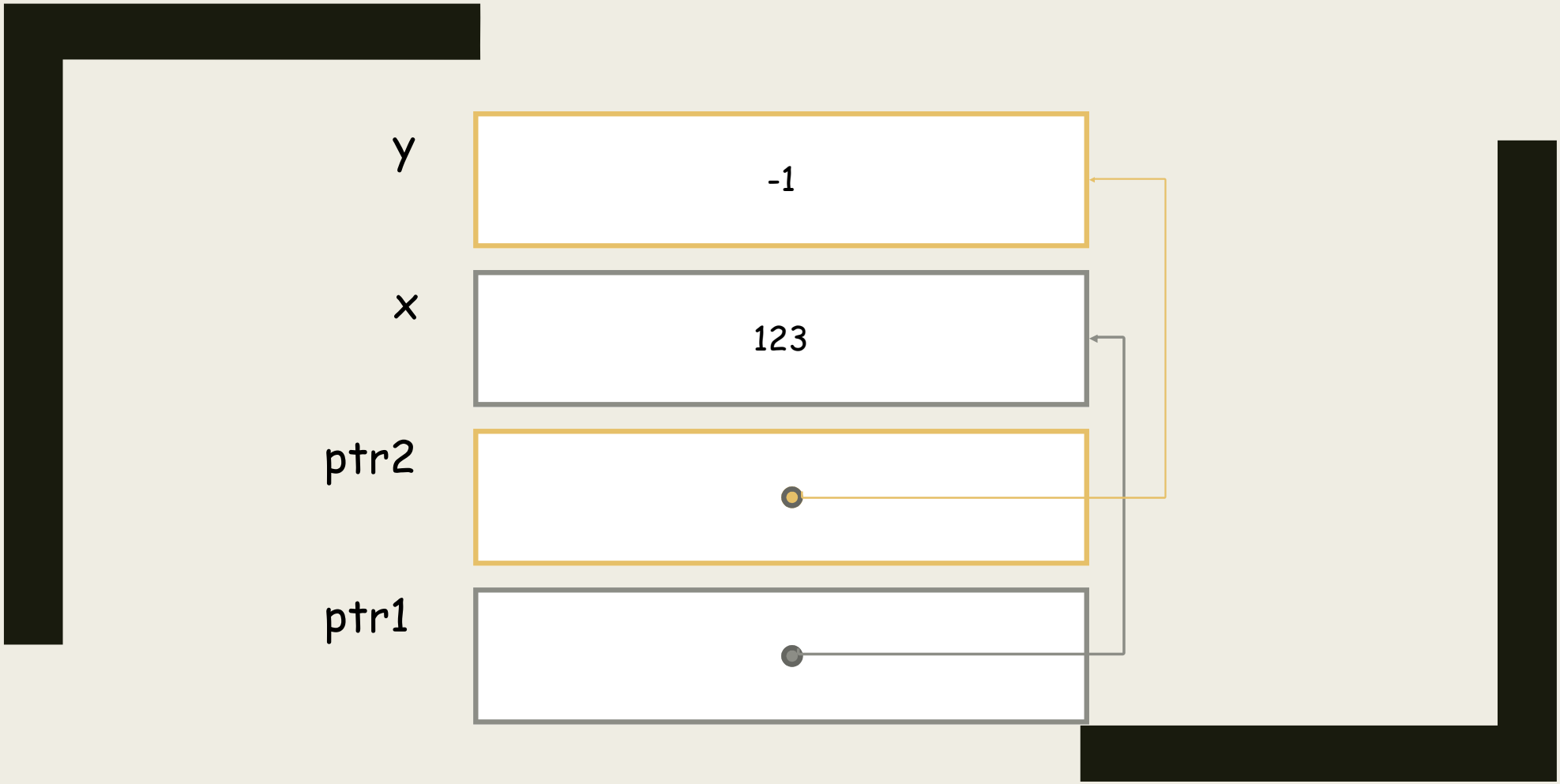

evantay@comp.nus.edu.sg

Problem Set 14.1

```
long *ptr1;
long *ptr2;
long x;
long y;

ptr1 = &x;
ptr2 = &y;

*ptr1 = 123;
*ptr2 = -1;
```

```
cs1010_println_long(x); // Prints: 123
cs1010_println_long(y); // Prints: -1
cs1010_println_long(*ptr1); // Prints: 123
cs1010_println_long(*ptr2);  // Prints: -1
```


evantay@comp.nus.edu.sg


y
-1
x
123
ptr2
ptr1

Problem Set 14.1

```
long *ptr1;
long *ptr2;
long x;
long y;
```

```
cs1010_println_long(x); // Prints: 123
cs1010_println_long(y); // Prints: -1
cs1010_println_long(*ptr1); // Prints: 123
cs1010_println_long(*ptr2);  // Prints: -1
```

```
ptr1 = &x;
ptr2 = &y;
```

```
ptr1 = ptr2;
*ptr1 = 1946;
```

```
*ptr1 = 123;
*ptr2 = -1;
```

```
cs1010_println_long(x); // Prints:
cs1010_println_long(y); // Prints:
cs1010_println_long(*ptr1); // Prints:
cs1010_println_long(*ptr2);  // Prints:
```

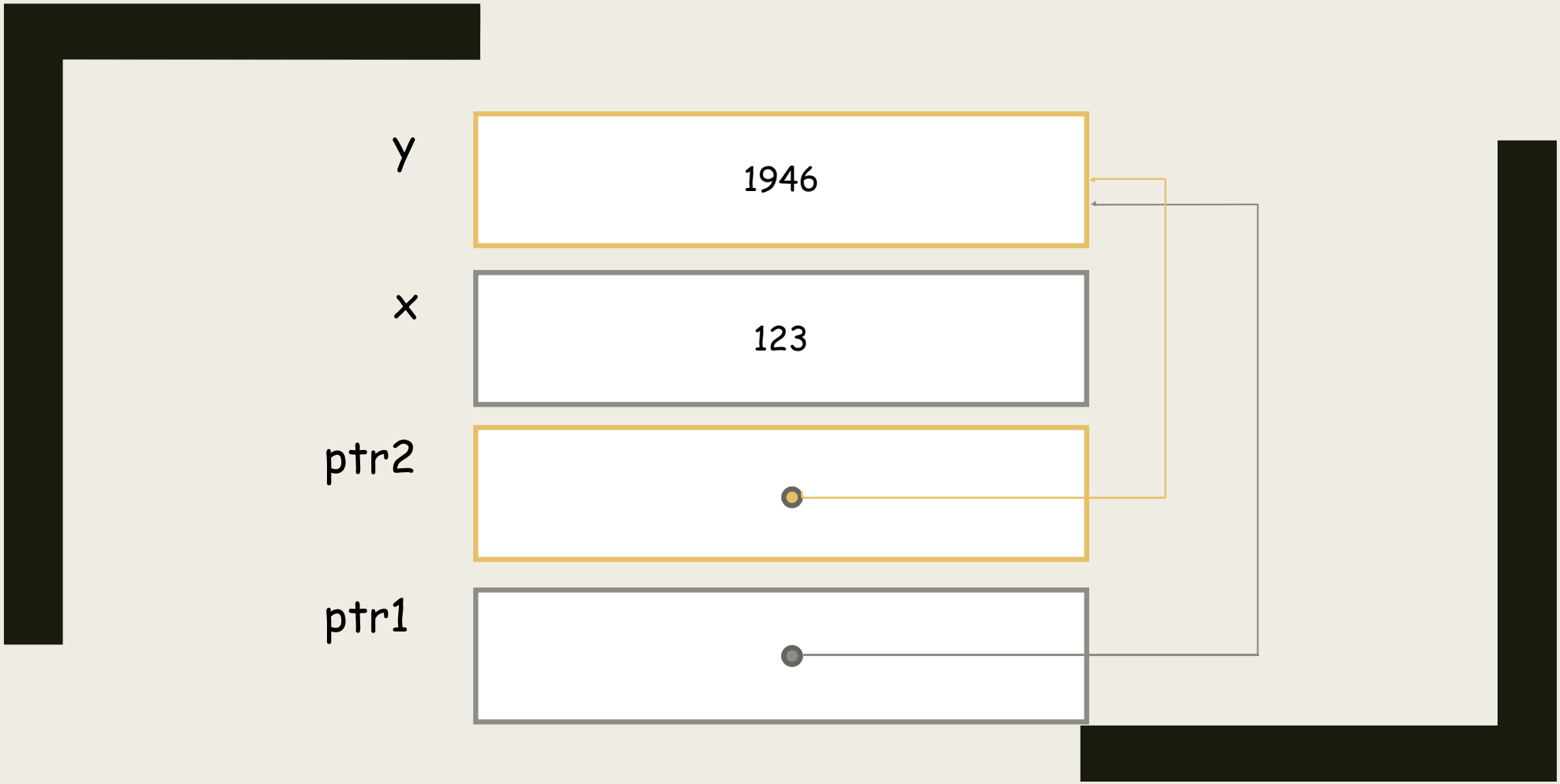

evantay@comp.nus.edu.sg

Problem Set 14.1

```
long *ptr1;
long *ptr2;
long x;
long y;

ptr1 = &x;
ptr2 = &y;

*ptr1 = 123;
*ptr2 = -1;
```

```
cs1010_println_long(x); // Prints: 123
cs1010_println_long(y); // Prints: -1
cs1010_println_long(*ptr1); // Prints: 123
cs1010_println_long(*ptr2);  // Prints: -1

ptr1 = ptr2;
*ptr1 = 1946;

cs1010_println_long(x); // Prints: 123
cs1010_println_long(y); // Prints: 1946
cs1010_println_long(*ptr1); // Prints: 1946
cs1010_println_long(*ptr2);  // Prints: 1946
```


evantay@comp.nus.edu.sg


y
1946
x
123
ptr2
ptr1

Problem Set 14.1

```
long *ptr1;
long *ptr2;
long x;
long y;

ptr1 = &x;
ptr2 = &y;

*ptr1 = 123;
*ptr2 = -1;
```

```
// ...

cs1010_println_long(x); // Prints: 123
cs1010_println_long(y); // Prints: 1946
cs1010_println_long(*ptr1); // Prints: 1946
cs1010_println_long(*ptr2);  // Prints: 1946

y = 10;

cs1010_println_long(x); // Prints:
cs1010_println_long(y); // Prints:
cs1010_println_long(*ptr1); // Prints:
cs1010_println_long(*ptr2);  // Prints:
```

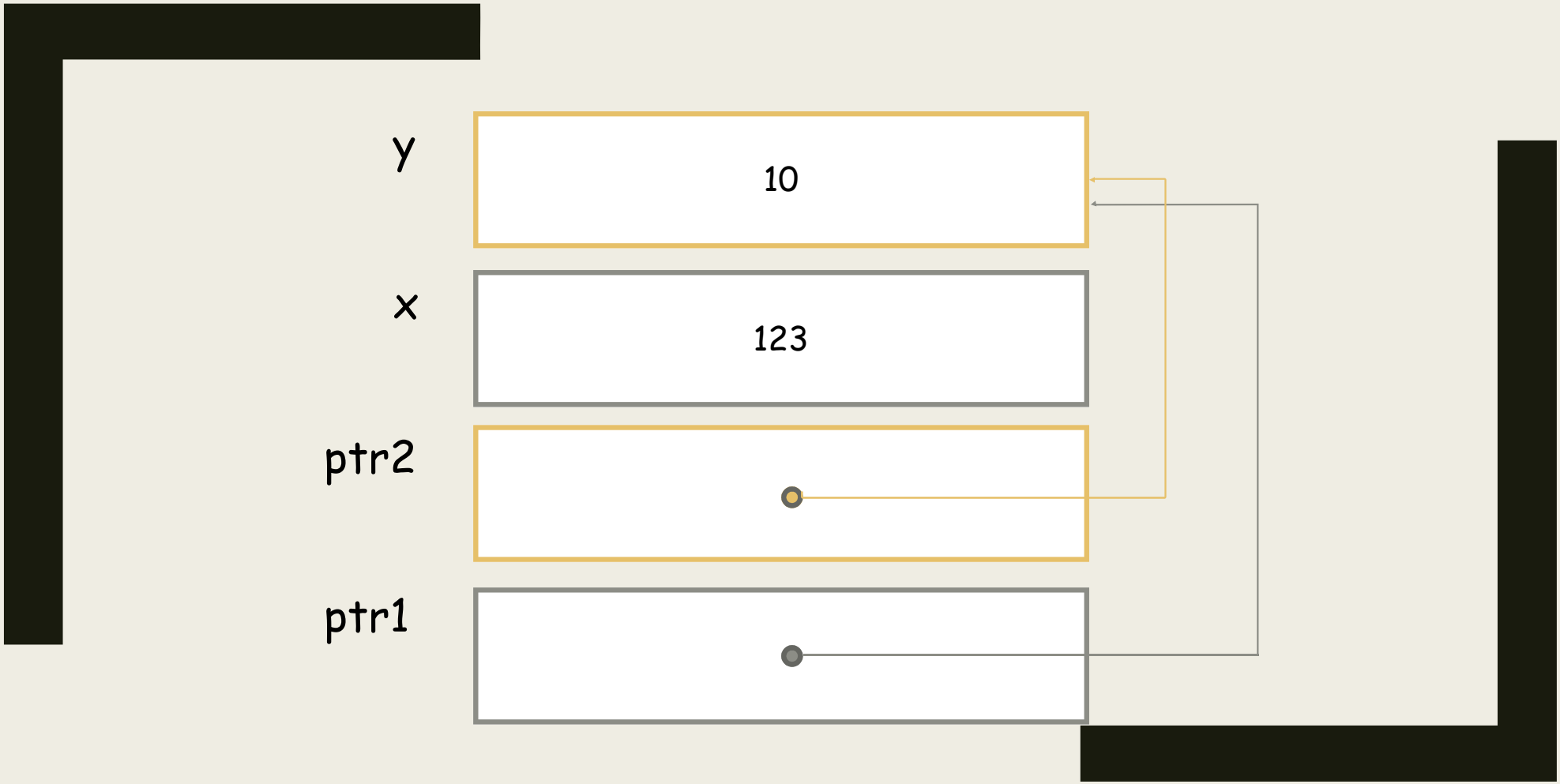

evantay@comp.nus.edu.sg

Problem Set 14.1

```
long *ptr1;
long *ptr2;
long x;
long y;

ptr1 = &x;
ptr2 = &y;

*ptr1 = 123;
*ptr2 = -1;
```

```
// ...

cs1010_println_long(x); // Prints: 123
cs1010_println_long(y); // Prints: 1946
cs1010_println_long(*ptr1); // Prints: 1946
cs1010_println_long(*ptr2);  // Prints: 1946

y = 10;

cs1010_println_long(x); // Prints: 123
cs1010_println_long(y); // Prints: 10
cs1010_println_long(*ptr1); // Prints: 10
cs1010_println_long(*ptr2);  // Prints: 10
```


evantay@comp.nus.edu.sg


y
10
x
123
ptr2
ptr1

PROBLEM SETS

13, 14, 15

Problem Set 14.2

```
double *addr_of(double x) {
  return &x;
}

int main() {
  double c = 0.0;
  double *ptr;
  ptr = addr_of(c);
  *ptr = 10;
}
```

```
double *triple_of(double x) {
  double triple = 3 * x;
  return &triple;
}
int main() {
  double *ptr;
  ptr = triple_of(10);
  cs1010_println_double(*ptr);
}
```


evantay@comp.nus.edu.sg

Problem Set 14.2

```
double *addr_of(double x) {
  return &x;
}

int main() {
  double c = 0.0;
  double *ptr;
  ptr = addr_of(c);
  *ptr = 10;
}
```

```
double *triple_of(double x) {
  double triple = 3 * x;
  return &triple;
}
int main() {
  double *ptr;
  ptr = triple_of(10);
  cs1010_println_double(*ptr);
}
```


evantay@comp.nus.edu.sg

PROBLEM SETS

13, 14, 15

Problem Set 15.1

- Write the function average that takes an array of k integers and k and returns the average of the k values in the array.

https://github.com/DigiPie/cs1010_tut_c09/blob/master/Tutorial_6/problem15_1.c

evantay@comp.nus.edu.sg

PROBLEM SETS

13, 14, 15

Problem Set 15.2 a)

```
int main() {
  long a = 0;
  printf("%ld\n", max(&a, 10));
  // What's wrong with this?
}
```


evantay@comp.nus.edu.sg

Problem Set 15.2 a)

```
int main() {
  long a = 0; // This only allocates one long value
  printf("%ld\n", max(&a, 10));
  // What's wrong with this?
  // Hence when max() attempts to access a[1], the program fails
}
```


evantay@comp.nus.edu.sg

Problem Set 15.2 b)

```
int main() {
  long a = 0; // This only allocates one long value
  printf("%ld\n", max(&a, 1));
  // What about this?
}
```


evantay@comp.nus.edu.sg

Problem Set 15.2 b)

```
int main() {
  long a = 0; // This only allocates one long value
  printf("%ld\n", max(&a, 1));
  // What about this? It works because only a[0] aka a* is accessed.
}
```


evantay@comp.nus.edu.sg

PROBLEM SETS

13, 14, 15

Problem Set 15.3

```
long max(long *list, long length) {
  long max_so_far;
  long *curr;
  max_so_far = *list;
  curr = list + 1;

  for (long i = 1; i != length; i += 1) {
    if (*curr > max_so_far) {
      max_so_far = *curr;
    }
    curr += 1;
  }
  return max_so_far;
}
```

1. ***curr** *is the similar to*
   - **list[i]**
2. **curr += 1** *is similar to*
   - **i += 1**
3. ***(curr + 5)** *is similar to*
   - **list[5]**

evantay@comp.nus.edu.sg

THE END

https://github.com/DigiPie/cs1010_tut_c09