CS1010

Evan Tay | *evantay@comp.nus.edu.sg*

https://github.com/DigiPie/cs1010_tut_c09

evantay@comp.nus.edu.sg

Today's plan

- Kahoot Quiz
- Recap
- Problem Sets 16, 17, 18, 19
- Consultation

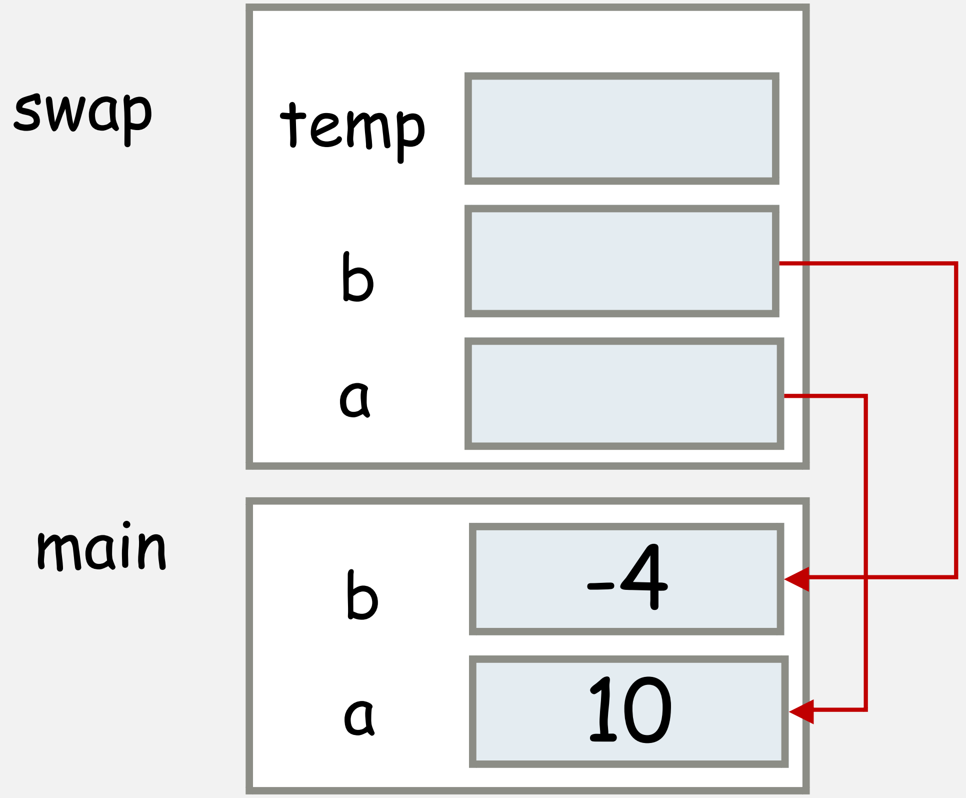
Kahoot!

RECAP

Strings, Call-by-reference, Heap and Multidimensional Array

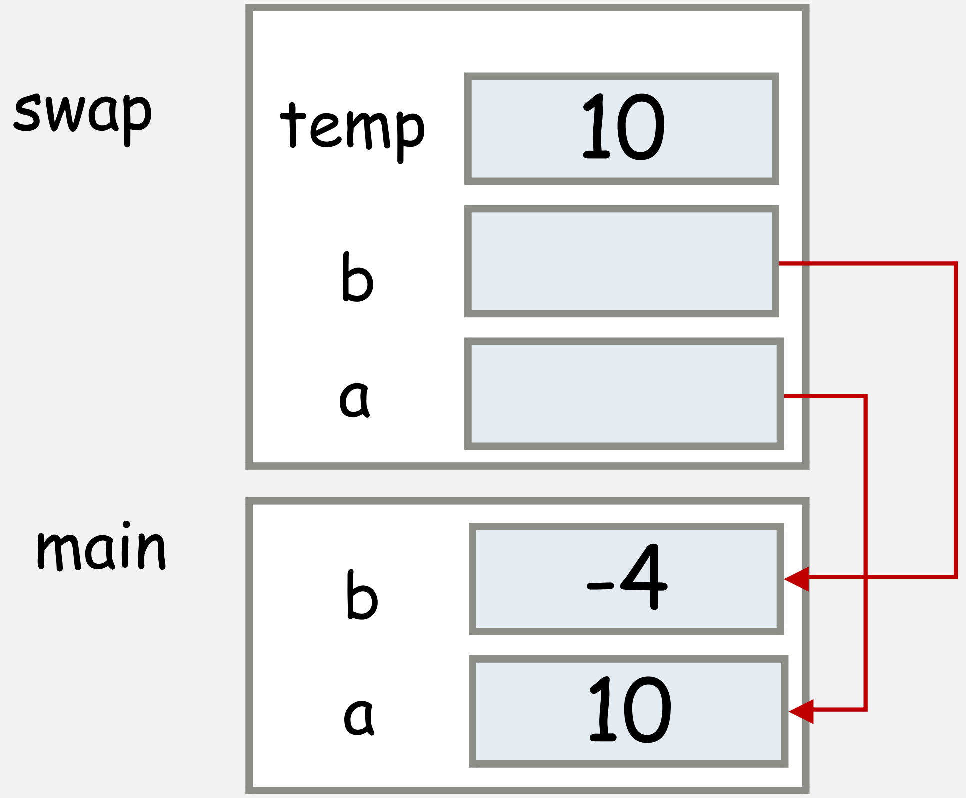
```
void swap(long *a, long *b) {
  long temp = *a;
  *a = *b;
  *b = temp;
}


long a = 10;
long b = -4;
swap(&a, &b);
```



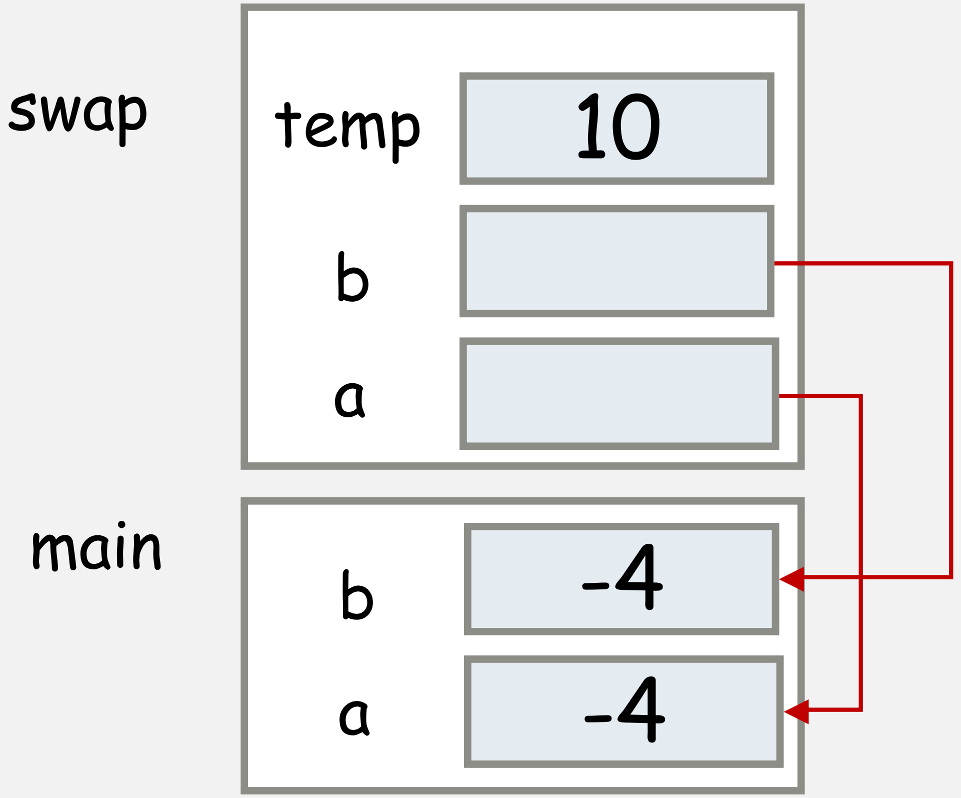
```
void swap(long *a, long *b) {
  long temp = *a;
  *a = *b;
  *b = temp;
}


long a = 10;
long b = -4;
swap(&a, &b);
```



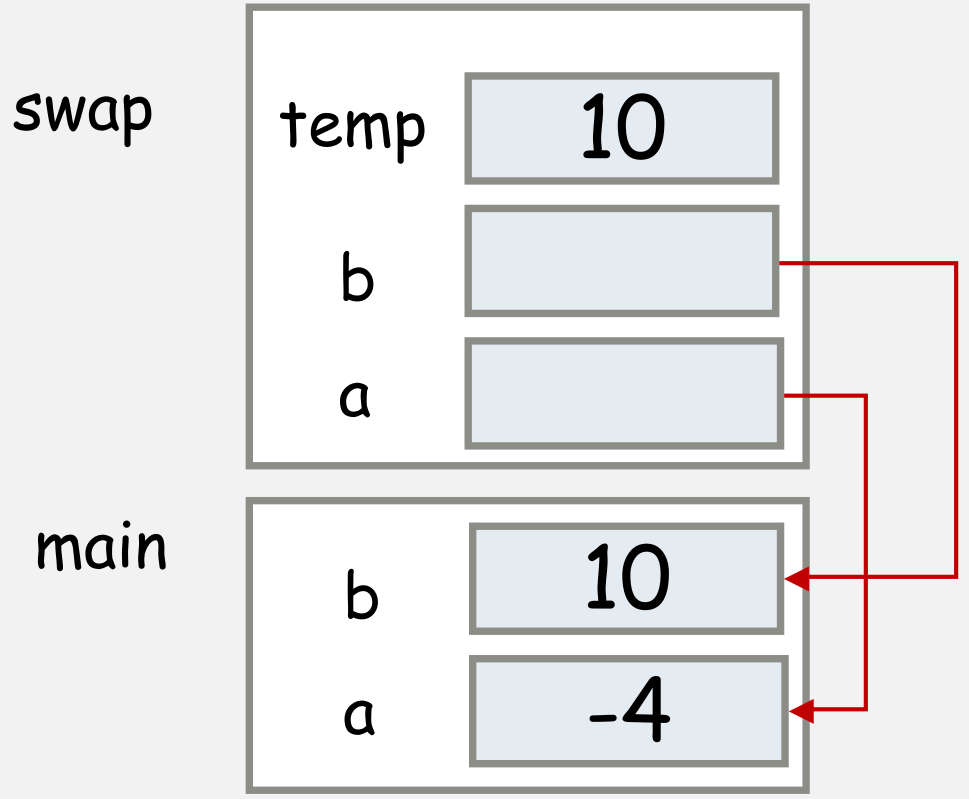
```
void swap(long *a, long *b) {
  long temp = *a;
  *a = *b;
  *b = temp;
}


long a = 10;
long b = -4;
swap(&a, &b);
```



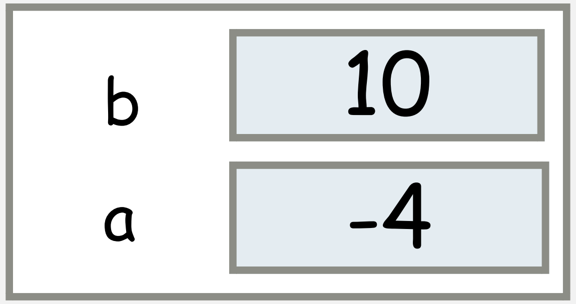
```
void swap(long *a, long *b) {
  long temp = *a;
  *a = *b;
  *b = temp;
}


long a = 10;
long b = -4;
swap(&a, &b);
```



```
void swap(long *a, long *b) {
  long temp = *a;
  *a = *b;
  *b = temp;
}


long a = 10;
long b = -4;
swap(&a, &b);
```



Why malloc

- We cannot use fixed length array unless we know for sure that the input size is limited, and we cannot use variable-length array on the stack, since we may get a segmentation fault if the array size is too big for the stack.
  - *The only viable solution is to allocate a variable-length array on the heap using malloc*

```
long a[3];
long (*p)[3]; // p is a pointer to an array
p = &a; // points p to the address of array a

if (p == a) {
  cs1010_println_string("same");
}

(*p)[2] = 1; // ok
(*a)[2] = 1; // error
```

PROBLEM SETS

16, 17, 18, 19

Problem Set 16.1

Write the following functions (without calling the standard C functions declared in <string.h> such as strlen, strcmp, strstr)…

https://github.com/DigiPie/cs1010_tut_c09/blob/master/Tutorial_7/problem16_1.c

PROBLEM SETS

16, 17, 18, 19

Problem Set 17.1

Complete the function find_min_max that takes in a length and an array containing long values of size length, and update the parameter min and max with the minimum and the maximum value from this array, respectively. Show how to call this function from main.

https://github.com/DigiPie/cs1010_tut_c09/blob/master/Tutorial_7/problem17_1.c

PROBLEM SETS

16, **17**, 18, 19

Problem Set 17.2

```
void foo(double *ptr, double trouble) {
  ptr = &trouble;
  *ptr = 10.0;
}
int main() {
  double *ptr;
  double x = -3.0;
  double y = 7.0;
  ptr = &y;
  foo(ptr, x);
  cs1010_println_double(x);
  cs1010_println_double(y);
}
```

What would be printed?

Problem Set 17.2

```
void foo(double *ptr, double trouble) {
  ptr = &trouble;
  *ptr = 10.0;
}
int main() {
  double *ptr;
  double x = -3.0;
  double y = 7.0;
  ptr = &y;
  foo(ptr, x);
  cs1010_println_double(x);
  cs1010_println_double(y);
}
```

What would be printed?

-3.0

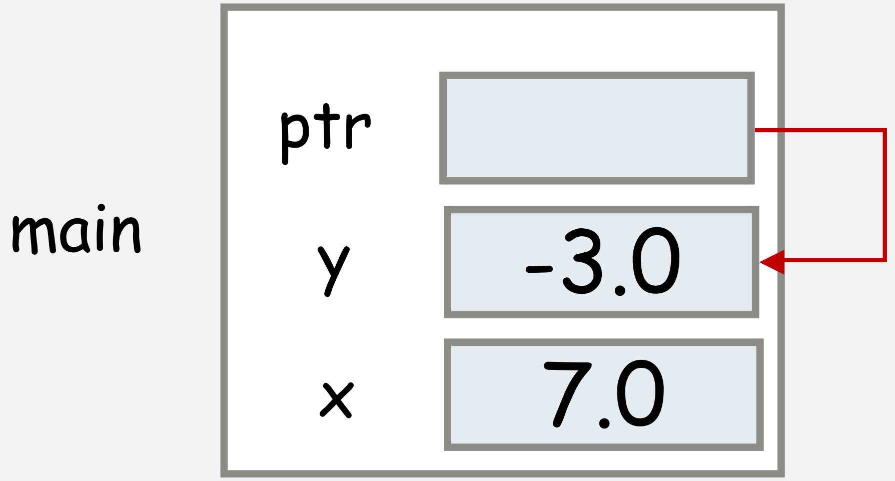
7.0

Because in foo,

*ptr and trouble are automatic variables.

Problem Set 17.2

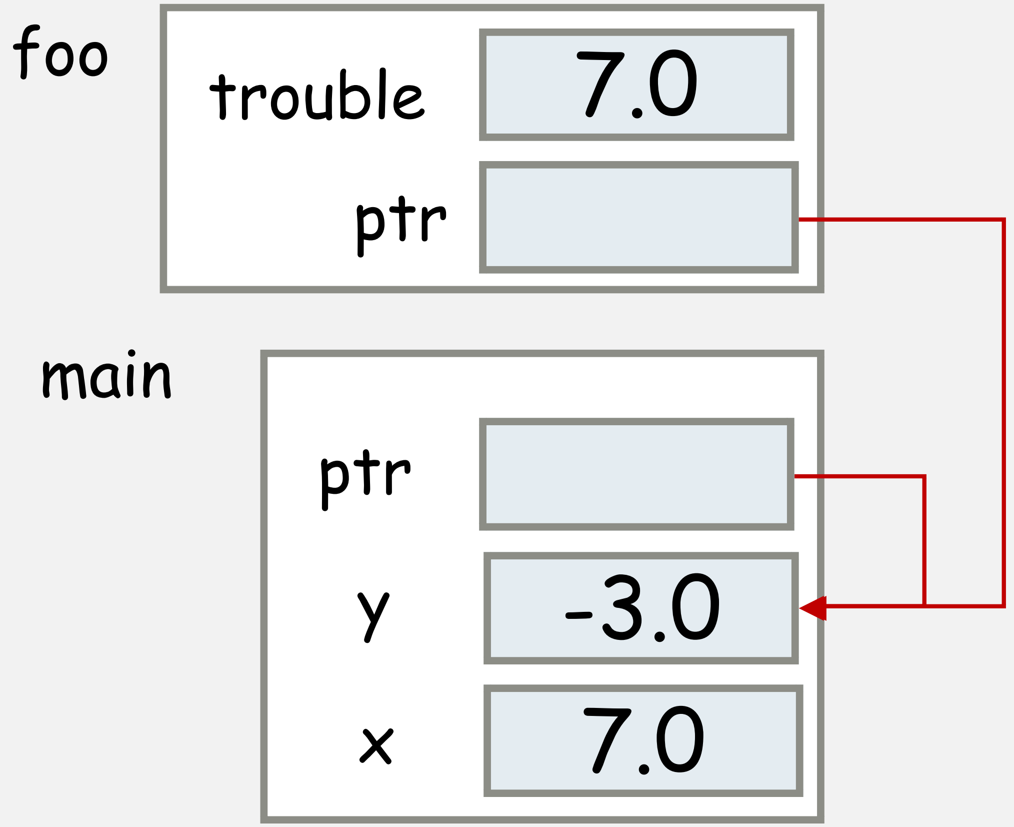
```
void foo(double *ptr, double trouble) {
  ptr = &trouble;
  *ptr = 10.0;
}
int main() {
  double *ptr;
  double x = -3.0;
  double y = 7.0;
  ptr = &y;
  foo(ptr, x);
  cs1010_println_double(x);
  cs1010_println_double(y);
}
```



evantay@comp.nus.edu.sg

Problem Set 17.2

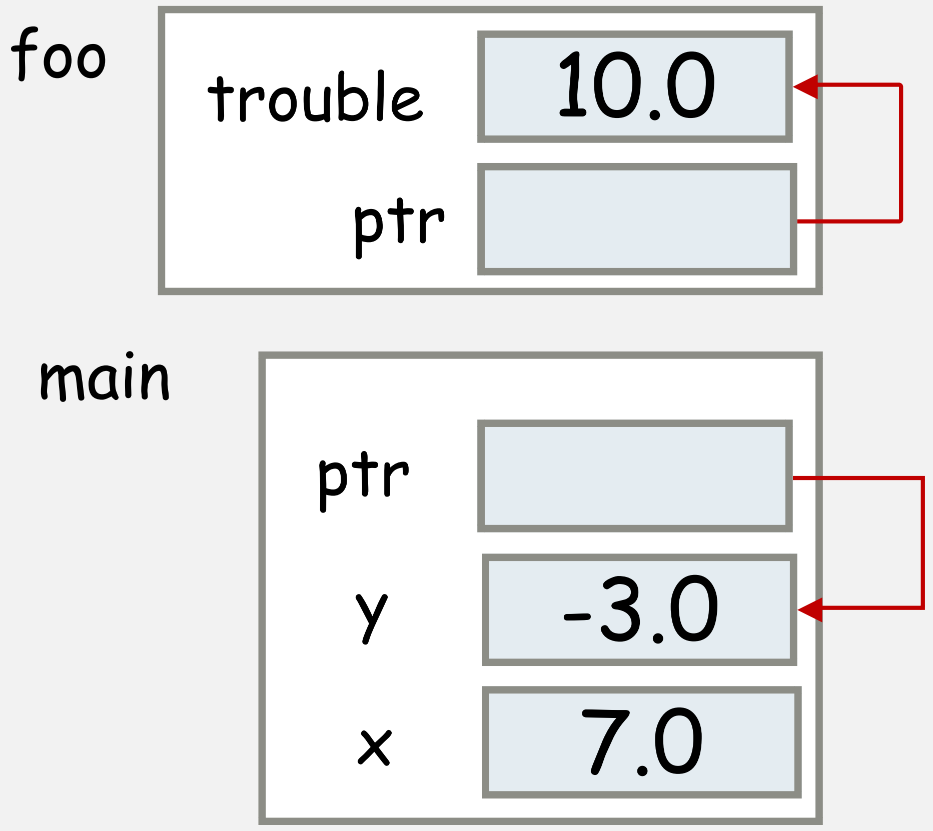
```
void foo(double *ptr, double trouble) {
  ptr = &trouble;
  *ptr = 10.0;
}
int main() {
  double *ptr;
  double x = -3.0;
  double y = 7.0;
  ptr = &y;
  foo(ptr, x);
  cs1010_println_double(x);
  cs1010_println_double(y);
}
```



evantay@comp.nus.edu.sg

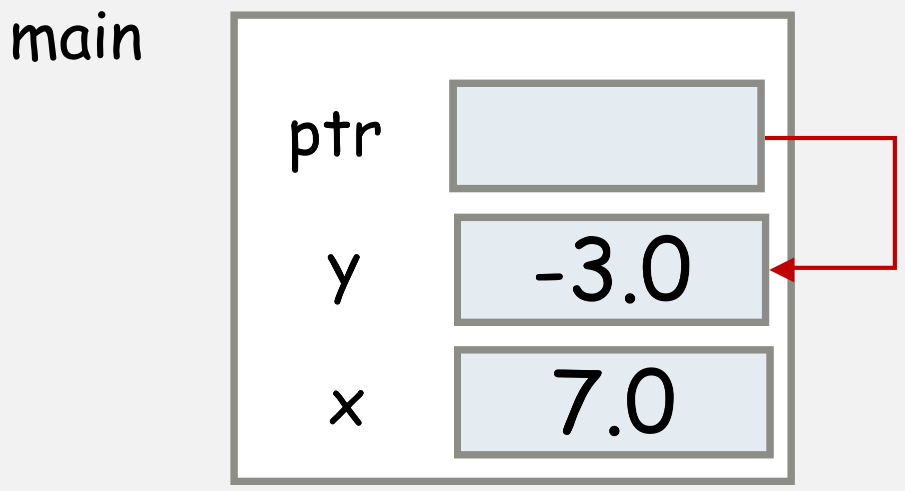
Problem Set 17.2

```
void foo(double *ptr, double trouble) {
  ptr = &trouble;
  *ptr = 10.0;
}
int main() {
  double *ptr;
  double x = -3.0;
  double y = 7.0;
  ptr = &y;
  foo(ptr, x);
  cs1010_println_double(x);
  cs1010_println_double(y);
}
```



Problem Set 17.2

```
void foo(double *ptr, double trouble) {
  ptr = &trouble;
  *ptr = 10.0;
}
int main() {
  double *ptr;
  double x = -3.0;
  double y = 7.0;
  ptr = &y;
  foo(ptr, x);
  cs1010_println_double(x);
  cs1010_println_double(y);
}
```



PROBLEM SETS

16, 17, **18**, 19

Problem Set 18.1

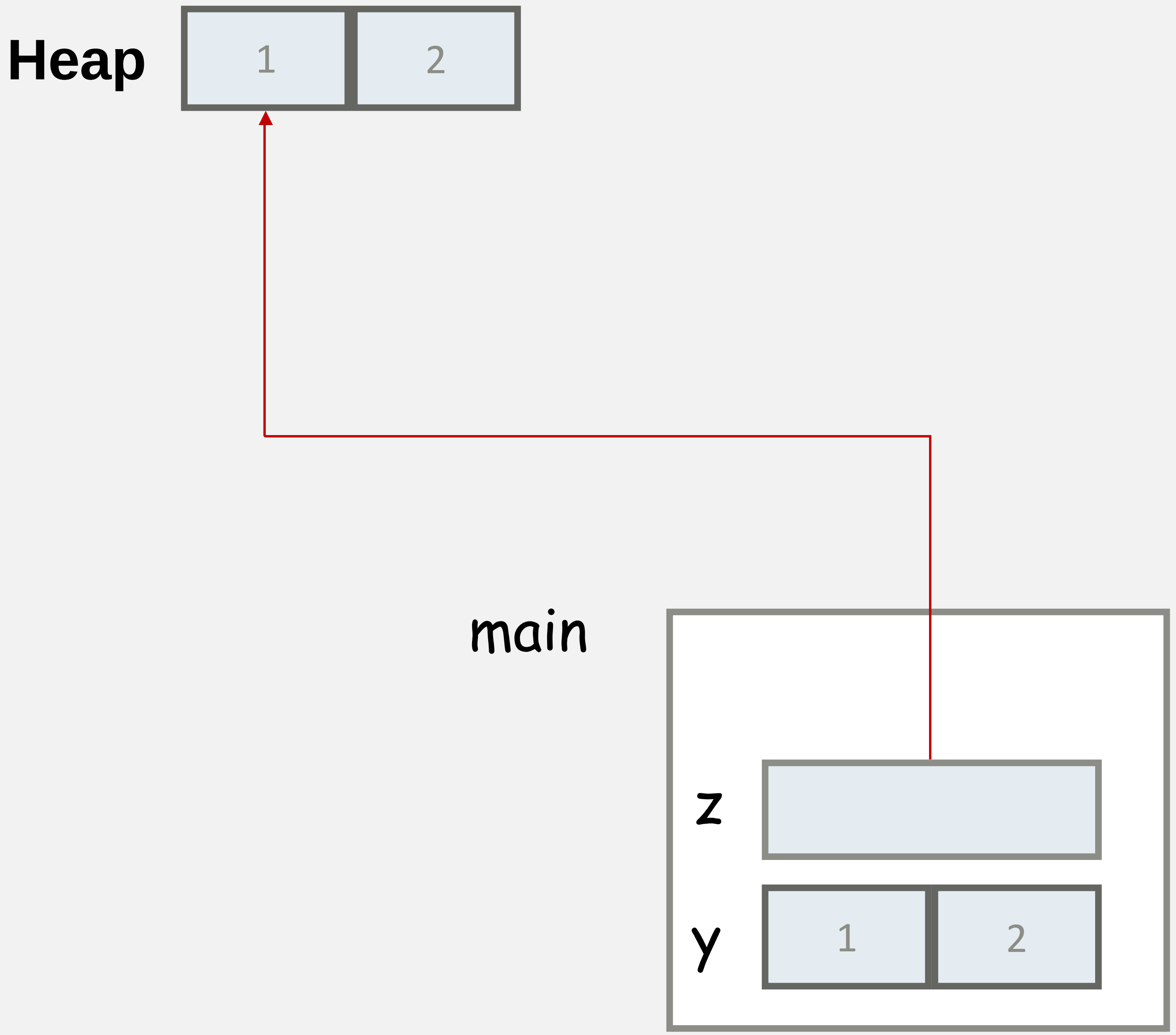
- Draw the call stack and the heap, showing what happened when we run the following code.

```
void foo(long *y, long *z) {
  y[0] = -7;
  y[1] = -8;
  z[0] = 4;
  z[1] = 5;
}
```

```
int main() {
  long y[2] = {1, 2};
  long *z = calloc(2, sizeof(long));
  z[0] = y[0];
  z[1] = y[1];
  foo(y, z);
}
```

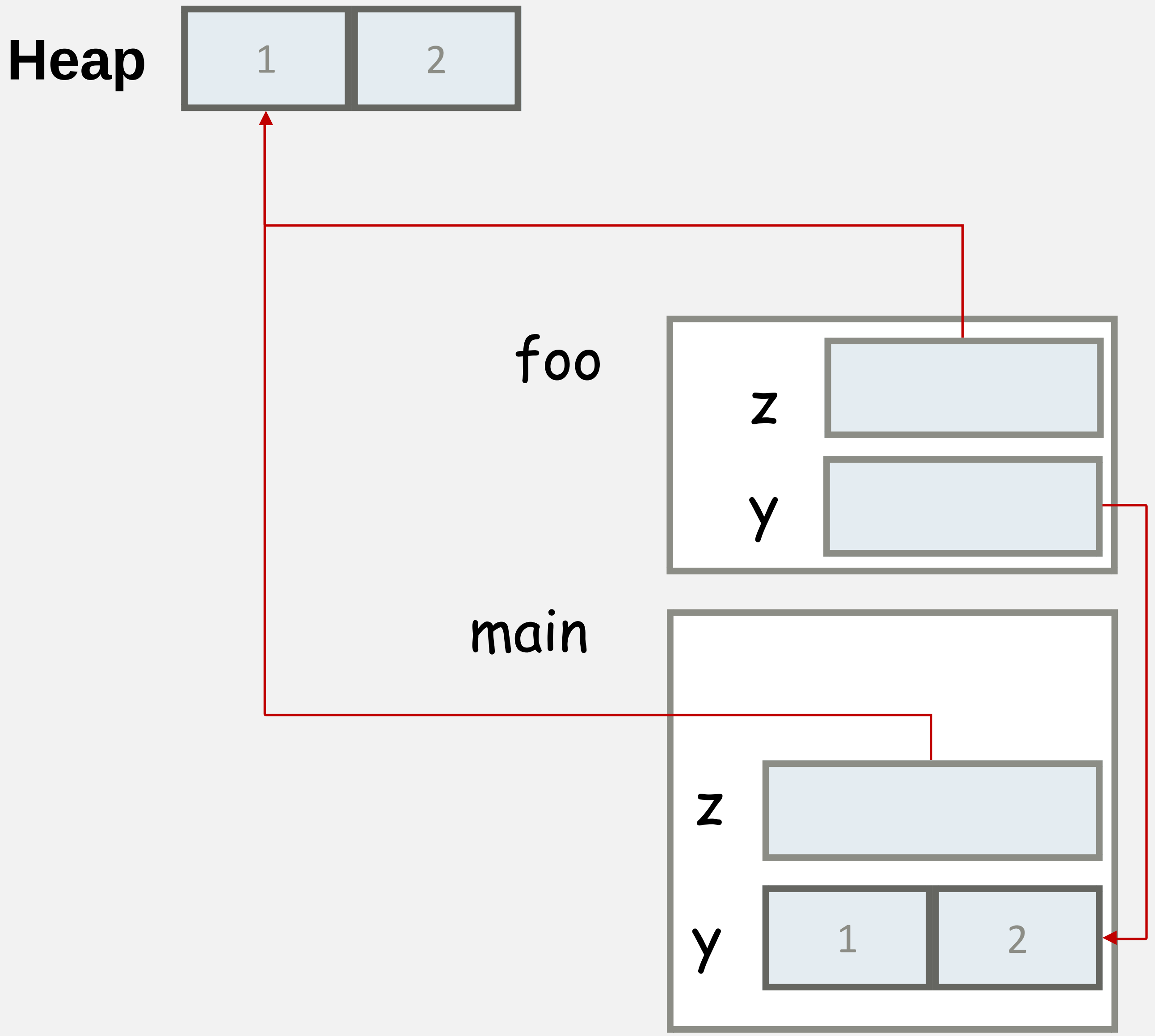
```
void foo(long *y, long *z) {
  y[0] = -7;
  y[1] = -8;
  z[0] = 4;
  z[1] = 5;
}

int main() {
  long y[2] = {1, 2};
  long *z = calloc(2, sizeof(long));
  z[0] = y[0];
  z[1] = y[1];
  foo(y, z);
}
```



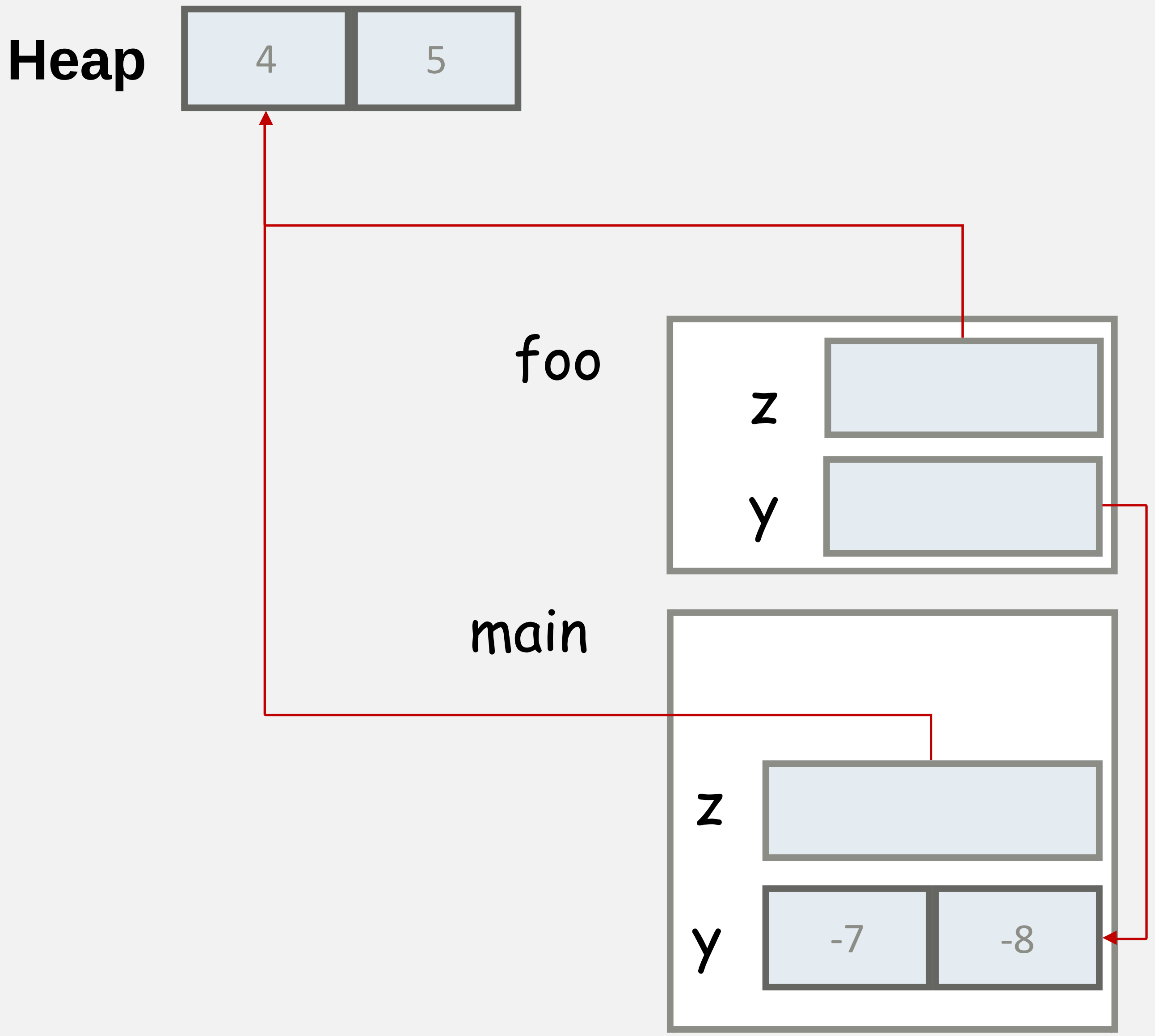
```
void foo(long *y, long *z) {
  y[0] = -7;
  y[1] = -8;
  z[0] = 4;
  z[1] = 5;
}

int main() {
  long y[2] = {1, 2};
  long *z = calloc(2, sizeof(long));
  z[0] = y[0];
  z[1] = y[1];
  foo(y, z);
}
```



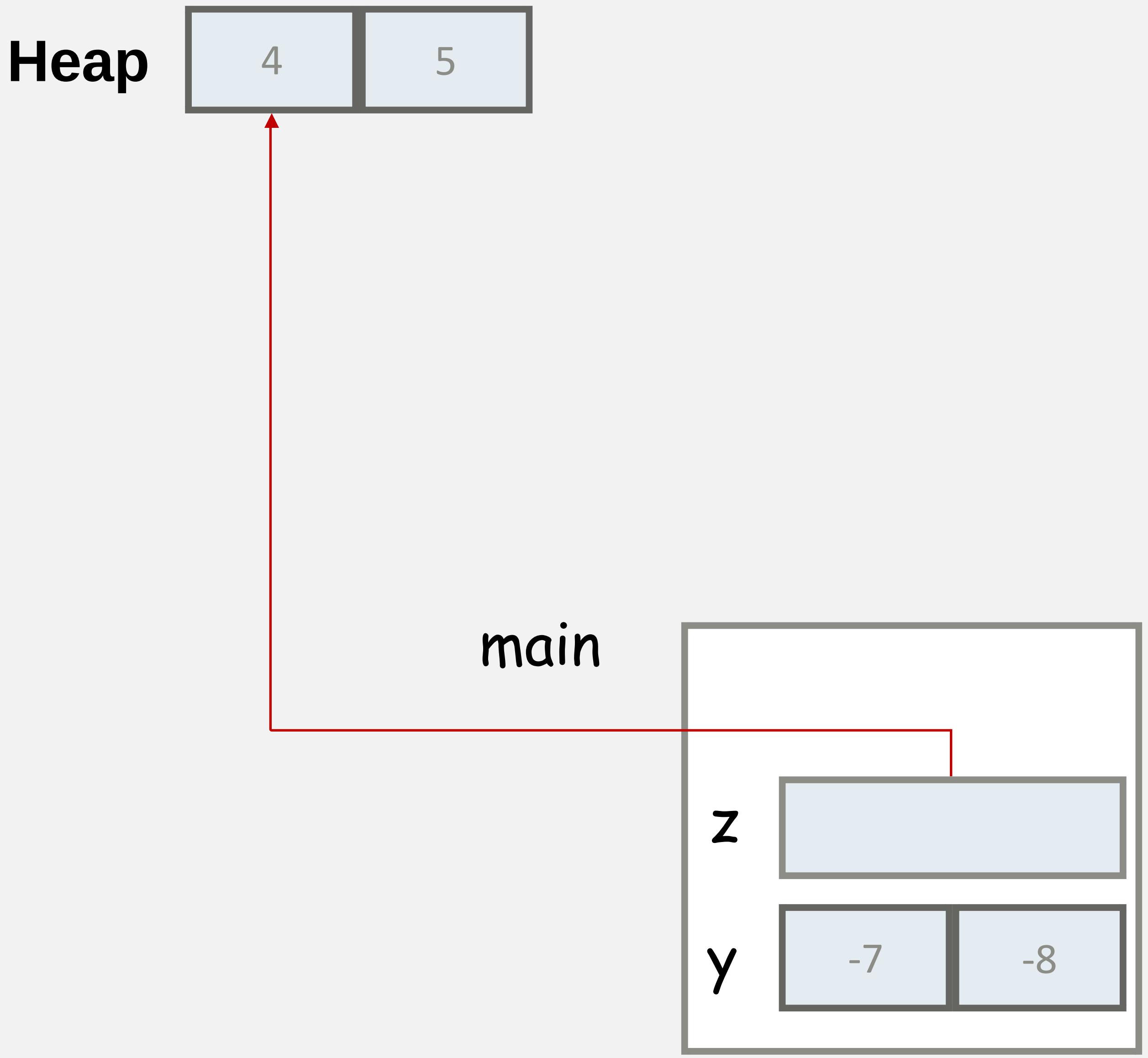
```
void foo(long *y, long *z) {
  y[0] = -7;
  y[1] = -8;
  z[0] = 4;
  z[1] = 5;
}

int main() {
  long y[2] = {1, 2};
  long *z = calloc(2, sizeof(long));
  z[0] = y[0];
  z[1] = y[1];
  foo(y, z);
}
```



```
void foo(long *y, long *z) {
  y[0] = -7;
  y[1] = -8;
  z[0] = 4;
  z[1] = 5;
}


int main() {
  long y[2] = {1, 2};
  long *z = calloc(2, sizeof(long));
  z[0] = y[0];
  z[1] = y[1];
  foo(y, z);
}
```



PROBLEM SETS

16, 17, **18**, 19

Problem Set 18.2

- When is realloc useful?

```
char *p = calloc(size, sizeof(char));
// some time later


if (i == size) {
  size *= 2;
  p = realloc(p, size); // double array size
}
```

PROBLEM SETS

16, 17, 18, **19**

Problem Set 19.1

a) Write a function add that performs 3x3 matrix addition. The function should operate on 3x3 matrices of long, takes in three parameters, the first two are the operands for addition and the third is the result.

b) Write a function multiply that performs 3x3 matrix multiplication. The function should operate on 3x3 matrices of long, takes in three parameters, the first two are the operands for multiplication and the third is the result.

Problem Set 19.1 a)

```
void add(long a[][3], long b[][3], long c[][3]) {
  for (int i = 0; i < 3; i += 1) {
    for (int j = 0; j < 3; j += 1) {
      c[i][j] = a[i][j] + b[i][j];
    }
  }
}
```

evantay@comp.nus.edu.sg

Problem Set 19.1 b)

```
long row_to_col(long a[][3], long b[][3], int row, int col) {
 long sum = 0;
 for (int i = 0; i < 3; i += 1) {
  sum += a[row][i] * b[i][col];
 }
 return sum;
}
```

```
void mul(long a[][3], long b[][3], long c[][3]) {
 for (int i = 0; i < 3; i += 1) {
  for (int j = 0; j < 3; j += 1) {
   c[i][j] = row_to_col(a, b, i, j);
  }
 }
}
```

PROBLEM SETS

16, 17, 18, 19

evantay@comp.nus.edu.sg

Problem Set 19.2

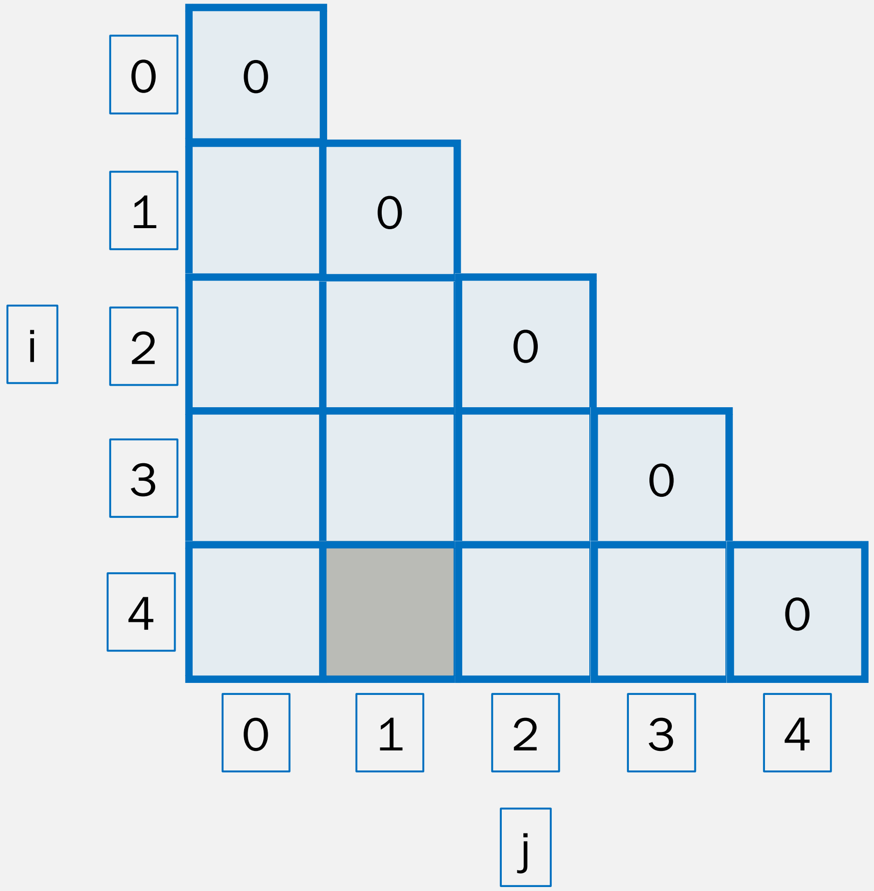
We need to represent the distance in km between every major cities in the world. Let's label every city with a number, ranging from 0 .. n−1, where n is the number of cities. The distance between city i and j is the same as the distance between city j and i. The distance can be represented with long.

Explain how you would represent this information using jagged two-dimensional array in C efficiently. We have information of a few thousand cities to store.

Explain how you would write `long dist(long **d, long i, long j)` to retrieve the distance between any two cities i and j.

evantay@comp.nus.edu.sg

Problem Set 19.2



- So, a matrix element d[i][j] is valid only if i >= j.
- dist(d, i, j) should return d[i][j] if i >= j, d[j][i] otherwise.
- d[i][i] should be 0

THE END

https://github.com/DigiPie/cs1010_tut_c09