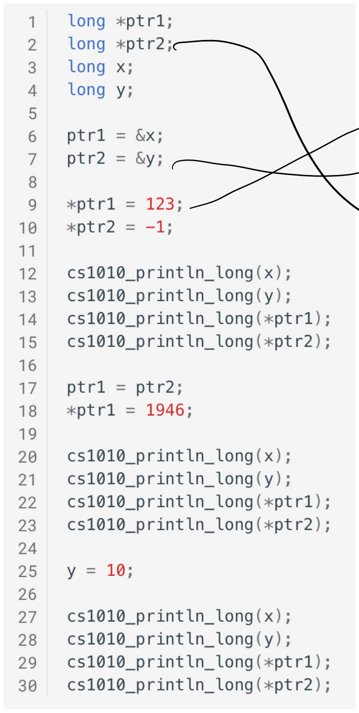


```
 1  long *ptr1;
 2  long *ptr2;
 3  long x;
 4  long y;
 5
 6  ptr1 = &x;
 7  ptr2 = &y;
 8
 9  *ptr1 = 123;
10  *ptr2 = -1;
11
12  cs1010_println_long(x);
13  cs1010_println_long(y);
14  cs1010_println_long(*ptr1);
15  cs1010_println_long(*ptr2);
16
17  ptr1 = ptr2;
18  *ptr1 = 1946;
19
20  cs1010_println_long(x);
21  cs1010_println_long(y);
22  cs1010_println_long(*ptr1);
23  cs1010_println_long(*ptr2);
24
25  y = 10;
26
27  cs1010_println_long(x);
28  cs1010_println_long(y);
29  cs1010_println_long(*ptr1);
30  cs1010_println_long(*ptr2);
```

what is * and & ?

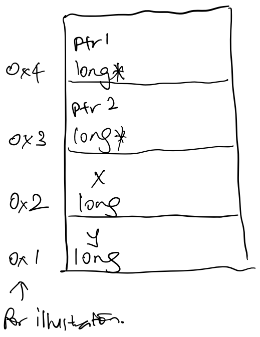
* deref

& ref.

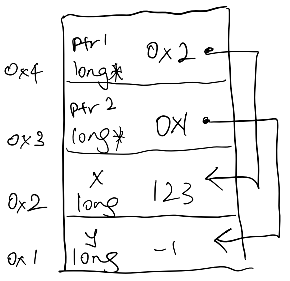
OR

multiplication / def. ptr

bitwise &



L11



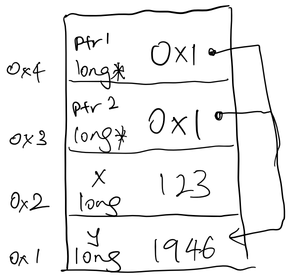
123

-1

123

-1

L18



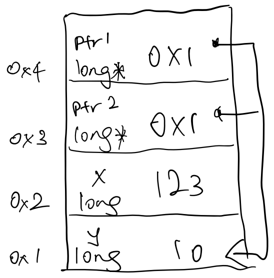
123

1946

1946

1946

L26



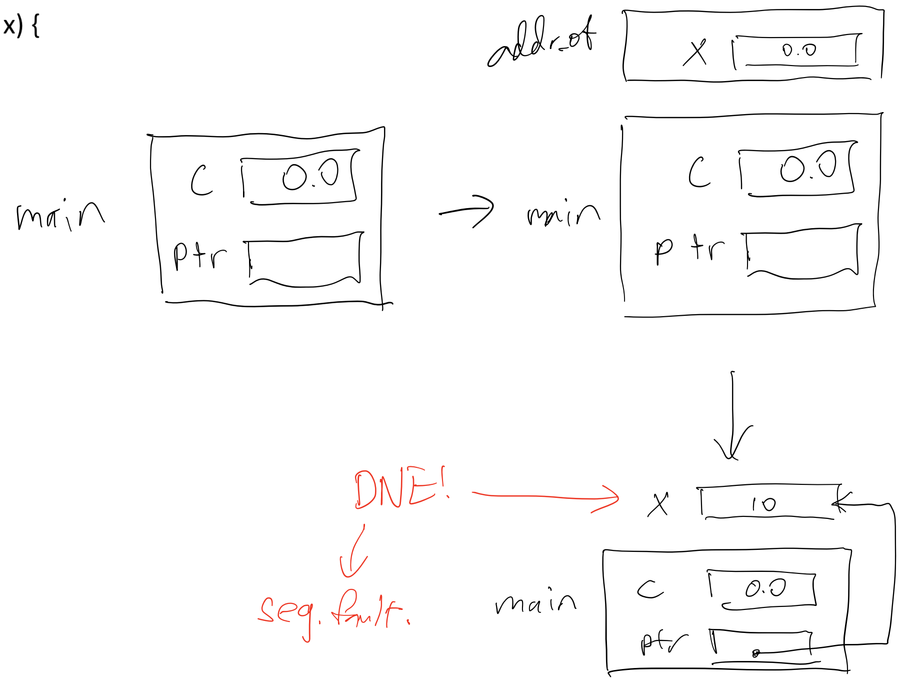
123

10

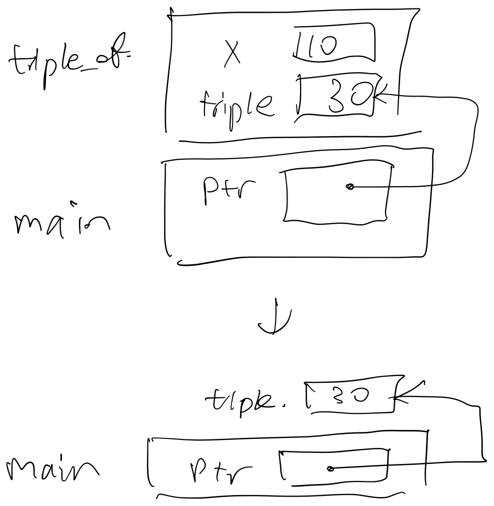
10

10

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



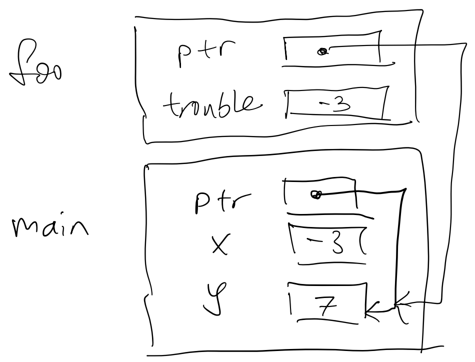
```
double *triple_of(double x) {
  double triple = 3 * x;
  return &triple;
}
int main() {
  double *ptr;
  ptr = triple_of(10);
  cs1010_println_double(*ptr);
}
```

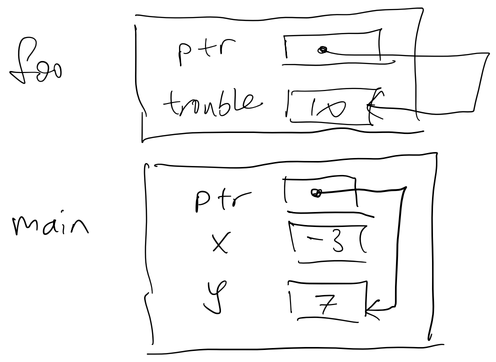


```
void foo(double *ptr, double trouble) {
  ptr = &trouble;
  *ptr = 10.0;
}
int main() {
  double *ptr;
  double x = -3.0;
  double y = 7.0;

  ptr = &y;
  foo(ptr, x);

  cs1010_println_double(x);
  cs1010_println_double(y);
}
```





Write a function that points the ptr in main to address of x

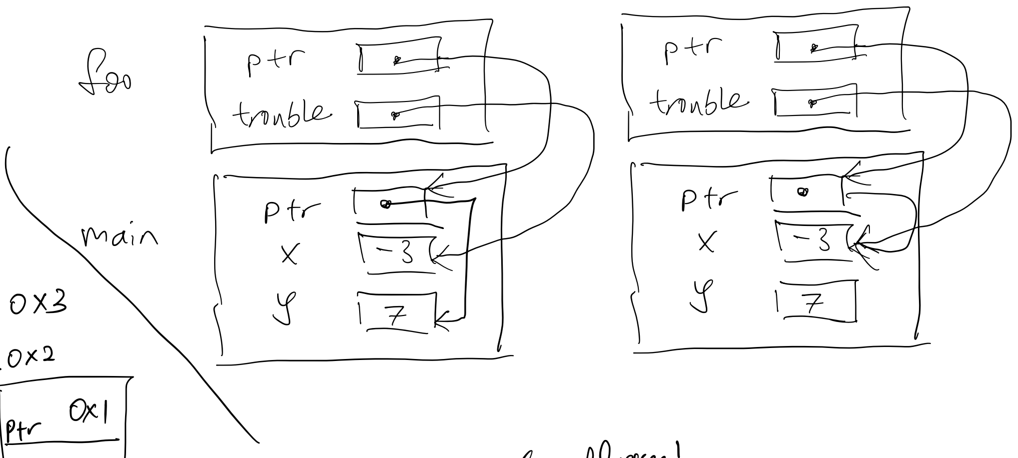
```
void foo(double **ptr, double *trouble) {
  *ptr = trouble;
}
int main() {
  double *ptr;
  double x = -3.0;
  double y = 7.0;

  ptr = &y;          0X3
  foo(&ptr, &x);     0x2

}
```

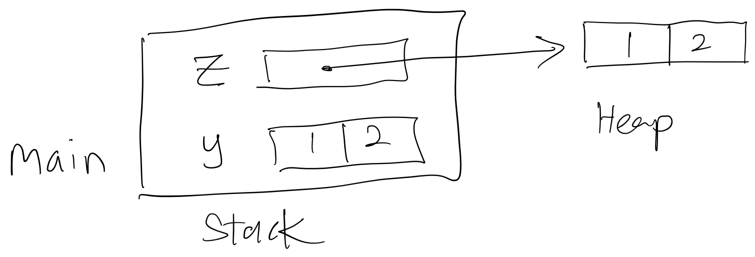
0X3	ptr	0x1
0x2	x	-3
0x1	y	7

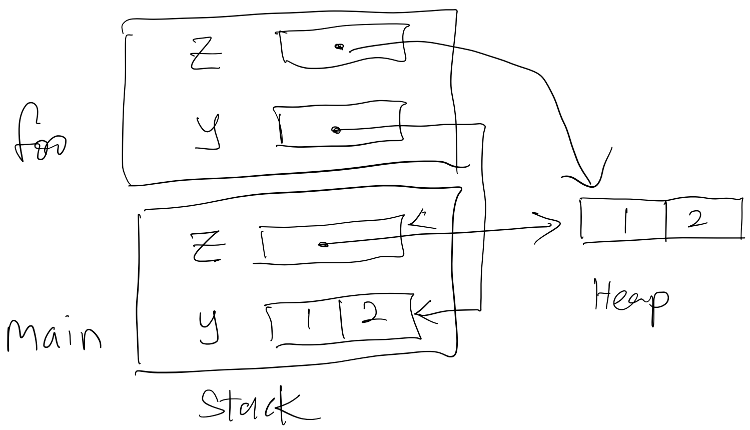
← visualize in terms of addresses!

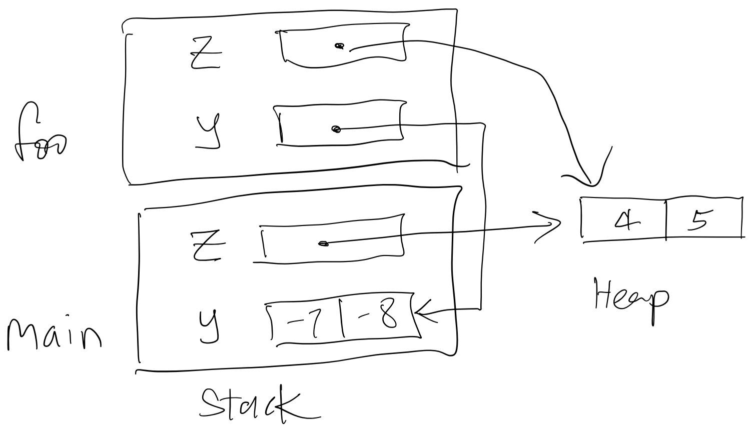


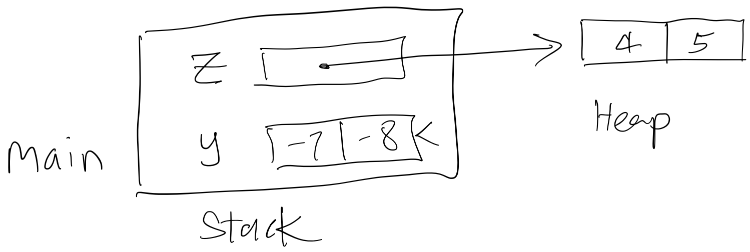
```
void foo(long *y, long *z) {
  y[0] = -7;
  y[1] = -8;
  z[0] = 4;
  z[1] = 5;
}

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


z
y
1
2
Main
Stack
1
2
Heap


foo
z
y
Main
z
y
1
2
Stack
1
2
Heap


foo
z
y
Main
z
y
-7
-8
Stack
4
5
Heap


z
y
-7
-8
Main
Stack
4
5
Heap