CSC209H Worksheet: Pointers

1. Each example below contains an independent code fragment. In each case there are variables `x` and `y` that are missing declaration statements. In the boxes to the right of the code write declaration statements so that the code fragment would compile and run without warnings or errors.

Code Fragment	Declaration for `x`	Declaration for `y`

`x = 10;`
`y = 'A';`
	x = 10	

`int age = 99;`
`x = &age;`
`y = *x;`
	int *x;	int y;

`double *p;`
`x = &p;`
`y = &x;`
	double **x;	double ***y;
		we have a pointer to a double pointer

`float f = 4.5;`
`float *p = &f;`
`x = &p;`
`y = **x;`
	float **x;	float y;
		(dereference twice)
		（ y = 4.5 ）

`char *result[2];` a date type
`x = result[0];` x is char *
`// some hidden code`
`result[0] = "read only";`
`y = x[0];`
char **x = results; (first element)
	you can use the name of a array as a pointer
	results is an array of two element
	resuts = [char *, char *]
	
	ans: char *x;
		ans: char y;

倒数第二行是把“read only”的address放到了result[0]那个pointer里，但x = chat*指向的还是原来的那个值。不是read only的

2. Trace through the following program either by hand to see why lying about your age doesn't last. Fix the program by changing the signature of the function to use a pointer.

```
#include <stdio.h>

void lie(int age) {
    printf("You are %d years old\n", age);
    age += 1;
    printf("You are %d years old\n", age);
}

int main() {

    int age = 18;
    lie(age);
    printf("But your age is still %d\n", age);

    return 0;
}
```

这个function会print 18 19 18

如果我们最后想print出19
我们要把line 2 变成 int *age
line 4 变成*age += 1;
line 3,5 变成printf("", *age);
line10变成 &age

CSC209H Worksheet: Pointers

3. Write a small program that allocates an array of integers in the main function and passes that array to a function. In the function, compute something using the elements of the array and return it. Compile and run your program to confirm you don't have syntax errors.

4. Add to the function in your program so that in addition to returning that value computed before, it modifies each element of the array. (Perhaps double them or square them or add some offset.) Print out the values of the array from main, after the function is called. Compile and run your code from the command line.

5. Change the function to take a second parameter that is an integer. Inside the function, perform the same operation on the individual integer as you performed on the array. Add code to main to declare and initialize the integer that is now the second function argument. Then print out the value of the integer after the function call.

6. With your partner answer the following questions: Does the integer change value inside the function? Do the array elements change value inside the function? Does the integer's value remain changed after the function returns? Do the array element values remain changed after the function returns?

7. Sketch a view of memory immediately before and after a function call that takes both an array of integers and a single integer as parameters.