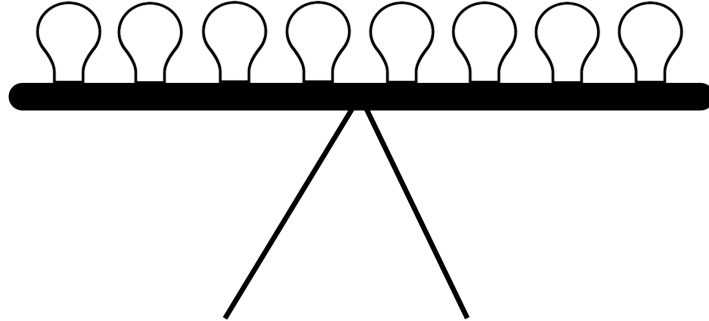


Binary Bulbs.

0. (1 point.) Consider the “binary bulbs” below that collectively represent a non-negative integer. Each of the bulbs represents a bit: a bulb that’s off is a 0, and a bulb that’s on is a 1. All of the bulbs are currently off. And the rightmost bulb represents the least significant bit (i.e., ones’ place).



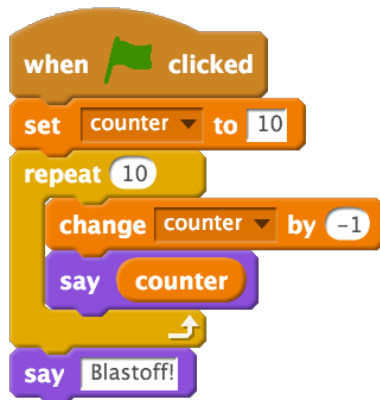
Suppose that you want these bulbs to represent the decimal integer 50. Turn on the requisite bulbs by marking any bulb that should become a 1, as by filling it in with your pen or pencil.

1. (1 point.) What is 0x25 in binary?

2. (1 point.) What is 0x33 in decimal?

Looping back to Scratch.

3. (4 points.) Consider the Scratch script below.



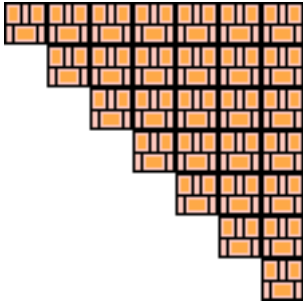
In the space below, complete the translation of this Scratch script to a C program in such a way that its output is equivalent. (Your program's structure needn't be equivalent.) Assume that **counter** is an `int`, that **say** is `printf`, and **change** means to increment or decrement. Output `\n` after each line of text.

```
#include <cs50.h>
#include <stdio.h>

int main(void)
{
```

Itsa Mario again.

4. (4 points.) Let's say that Mario comes across the “half-pyramid” of blocks like in the screenshot below. Good luck jumping onto that one!



Complete the implementation of the program below in such a way that it recreates this particular half-pyramid using hashes (#) for blocks. No need for user input; you may hard-code the half-pyramid's height (7) into your program. However, your algorithm should work for any reasonable height of pyramid!

```
#include <cs50.h>
#include <stdio.h>
```

```
int main(void)
{
```

Quick questions

For each of the questions below, circle the correct answer.

5. (1 point.) What will be the value of `c` after executing the code fragment below?

```
int a = 1;
int b = 2;
int c = 3;
if (a < b)
{
    c = b;
}
else if (a > b)
{
    c = a;
}
a = 4;
b = 5;
```

6. (1 point.) What will be printed if we run the following code?

```
string s = "Zamyla";
printf("%d\n", s[5]);
```

7. (1 point.) What will be printed if we run the following code?

```
char s[] = "Quinten";

for(int i = 0; i < 8; i++)
{
    if(s[i] == 'Q')
    {
        s[i] = 'X';
    }
}

printf("%s\n", s);
```

8. (1 point.) What is the result of the following code?

```
char s[] = "Hi\0Jill";
```

- a. The two characters 'H' and 'i' are stored in `s`; `strlen(s) == 2`
- b. The seven characters 'H', 'i', '\0', 'J', 'i', 'l' and 'l' are stored in `s`; `strlen(s) == 7`
- c. The eight characters 'H', 'i', '\0', 'J', 'i', 'l', 'l' and '\0' are stored in `s`; `strlen(s) == 7`
- d. The eight characters 'H', 'i', '\0', 'J', 'i', 'l', 'l' and '\0' are stored in `s`; `strlen(s) == 2`

9. (1 point.) What will be printed if we run the following code?

```
#include <stdio.h>
#include <cs50.h>

void print_name(string name);

int main(void)
{
    string input = "Julia";
    print_name(input);
    input = "Emma";
}

void print_name(string name)
{
    printf("hello, %s\n", name);
}
```

- a. 1
- b. 2
- c. 3
- d. 4

- a. hello, name
- b. hello, Julia
- c. hello, Emma
- d. nothing, compile error

- a. 97
- b. \0
- c. nothing, error
- d. a

10. (1 point.) Fill in the dots to make sure that this code fragment produces no compiler errors.

```
... add_this(int a, int b)
{
    return a / b;
}
```

- a. Quinten
- b. X
- c. nothing, error
- d. Xuinten

- a. int
- b. float
- c. any of the above
- d. nothing

11. (1 point.) Upon compiling `hello.c`, we get the following error. What would be the best solution?

```
hello.c:3:5: error: implicitly declaring library function 'printf'
```

- a. adding a prototype to `hello.c` for the function `printf`
- b. including `stdio.h` in `hello.c`
- c. changing the compiler flags to include the `stdio` library
- d. using `get_string` instead of `printf`

Bubbling up and down

Recall the procedure for bubble sort: "We move down the list from left to right and compare each pair of numbers. If they are out of order, then we swap them. The list isn't sorted yet but the highest number is now on the right, and the other numbers are slightly closer to where they should be. We repeat the procedure until the list is sorted."

12. (4 points.) Perform that procedure on the following array, until sorted. Show your work line by line.

4	2	6	8	1

13. (1 point.) Given some input of n numbers, what would be the worst-case running time of this algorithm?
- a. $O(1)$
 - b. $O(n)$
 - c. $O(\log n)$
 - d. $O(n \log n)$
 - e. $O(n^2)$
14. (1 point.) Given some input of n numbers, what would be the best-case running time of this algorithm?
- a. $\Omega(1)$
 - b. $\Omega(n)$
 - c. $\Omega(\log n)$
 - d. $\Omega(n \log n)$
 - e. $\Omega(n^2)$