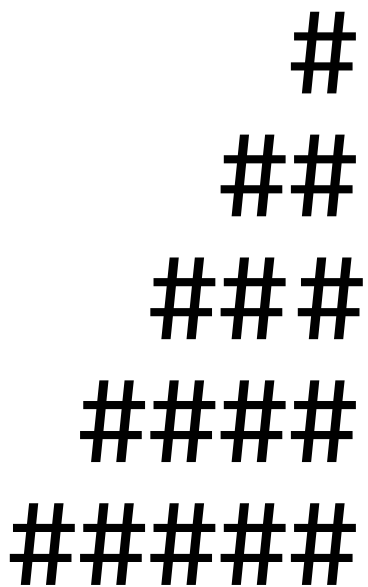


This is CS50.

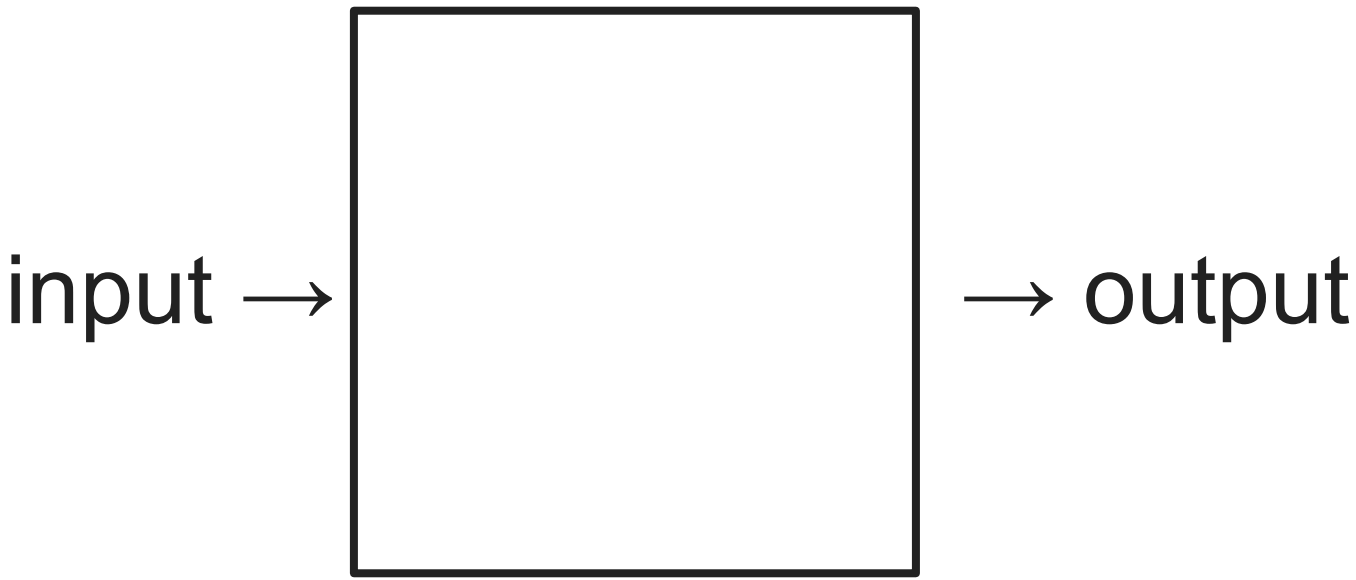
Think.
Pair.
Share.

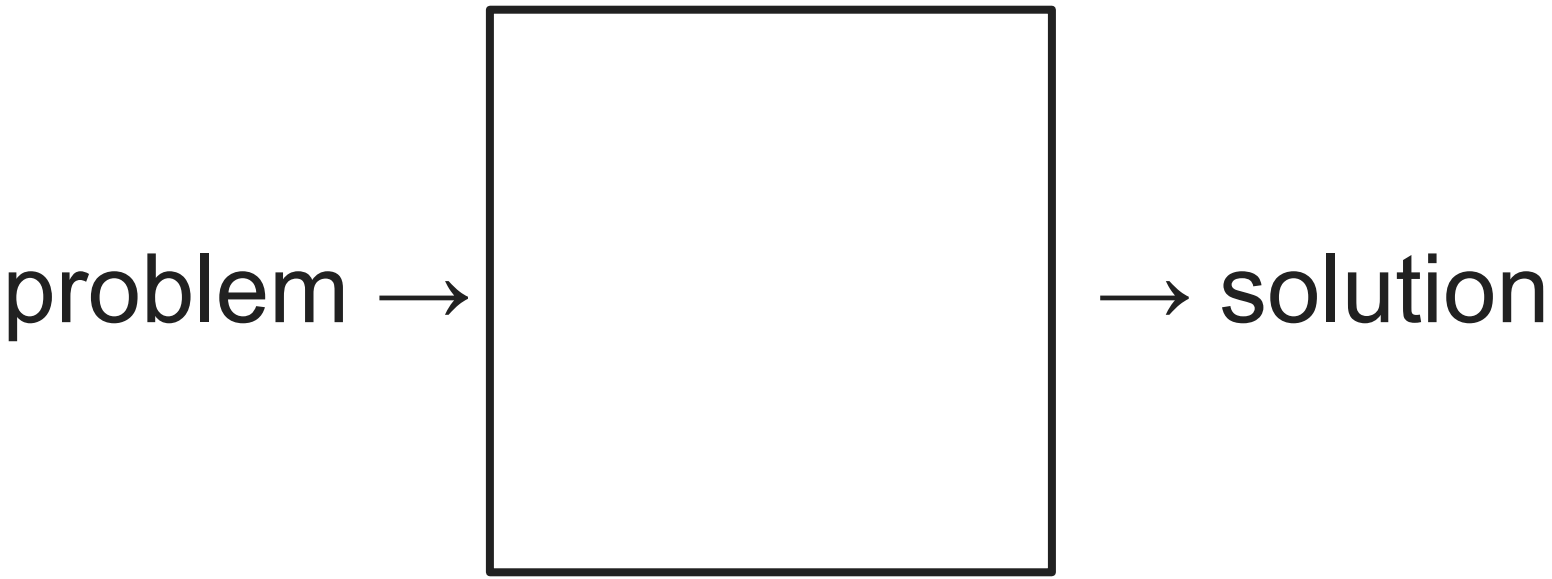
cs50.ly/questions

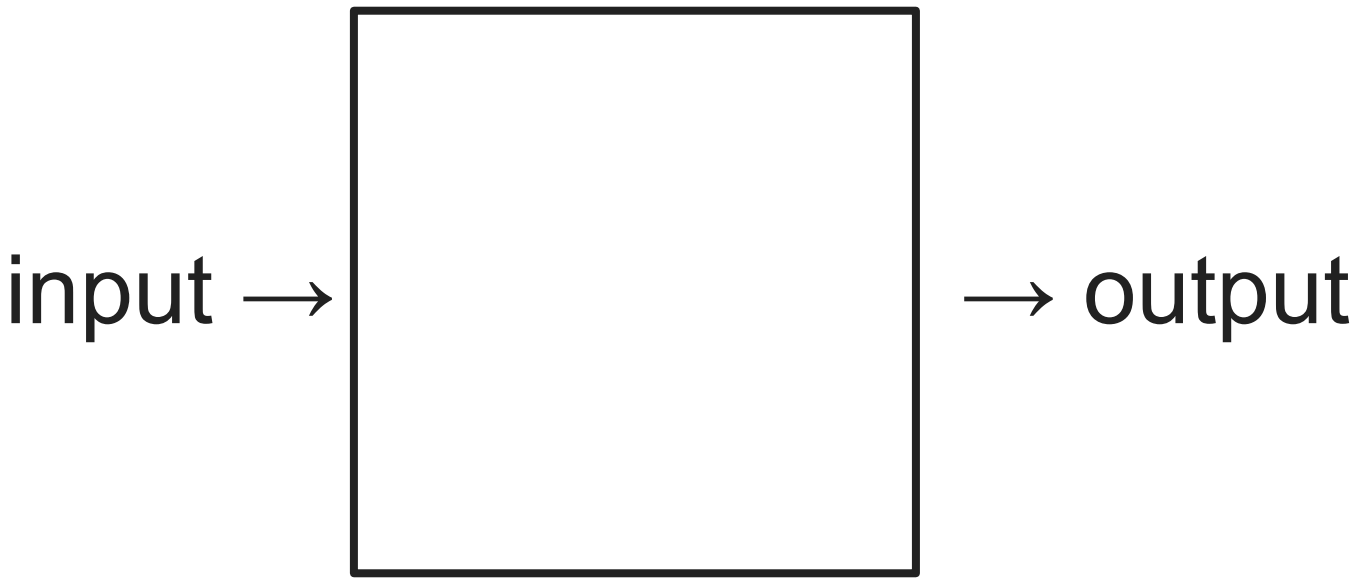
- What are the steps involved in **compilation**?
  - What is the role of **trust** in computer science?
- When should we use **arrays**?
- What are **strings**, really?
- What's the point of **command-line arguments**?

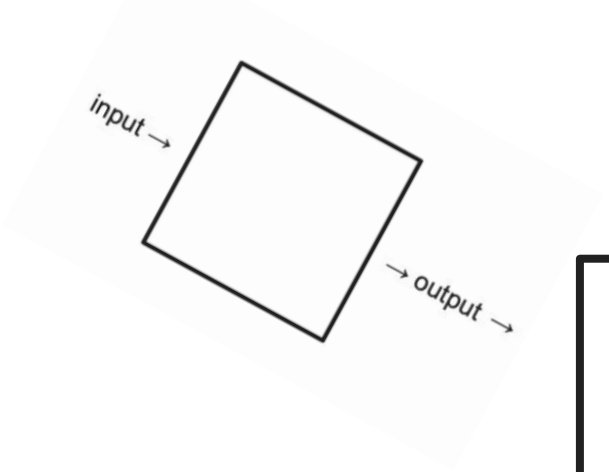
- What are the steps involved in **compilation**?
  - What is the role of **trust** in computer science?
- When should we use **arrays**?
- What are **strings**, really?
- What's the point of **command-line arguments**?
- What makes for good **design**?

- What are the steps involved in **compilation**?
  - What is the role of **trust** in computer science?
- When should we use **arrays**?
- What are **strings**, really?
- What's the point of **command-line arguments**?
- What makes for good **design**?

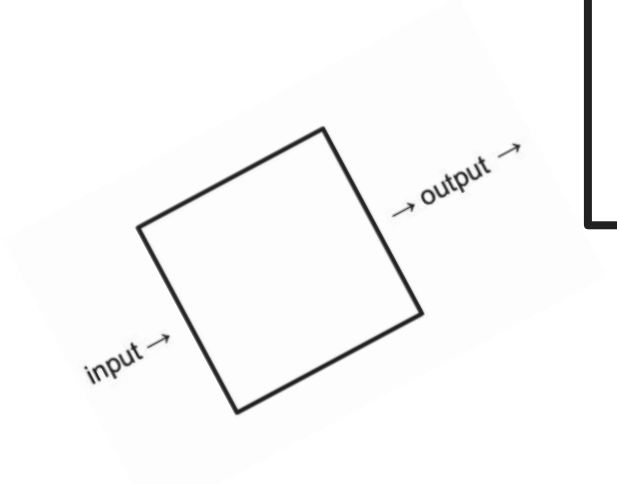

input →
→ output

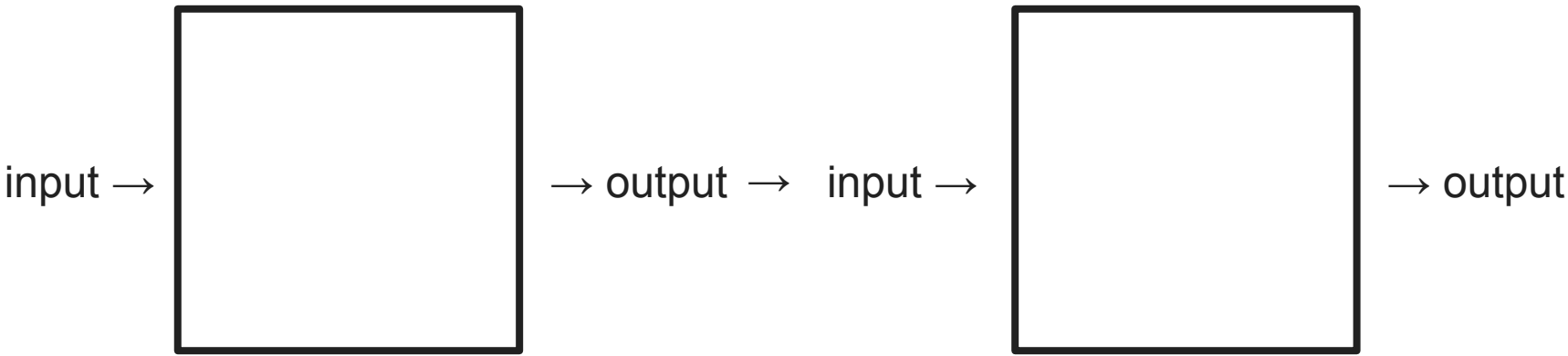

problem →
→ solution


input →
→ output



input → → output




input →
→ output →
input →
→ output

```
int main(void)
{
    print("Hello");
}
```

```
...
main:                                   #
@main
    .cfi_startproc
# BB#0:
    push    %rbp
.Ltmp0:
    .cfi_def_cfa_offset 16
.Ltmp1:
    .cfi_offset %rbp, -16
    movq    %rsp, %rbp
.Ltmp2:
    .cfi_def_cfa_register %rbp
```

```
01111111010001010100110001000110
00000010000000010000000100000000
00000000000000000000000000000000
00000000000000000000000000000000
00000001000000000011111000000000
00000001000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
10100000000000100000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
01000000000000000000000000000000
00000000000000000100000000000000
00001010000000000000000100000000
01010101010010001000100111100101
01001000100000111110110000010000
00110001110000001000100111000111
01001000101111100000000000000000
00000000000000000000000000000000
00000000000000001011000000000000
11101000000000000000000000000000
00000000010010001011111100000000
00000000000000000000000000000000
00000000000000000000000001001000
```

Snake

Arrays

good morning ☀️ it's a new day!
how was last night? 👀

how many hours did you sleep last night? *

7.5

how many times did you snooze your alarm this morning? * (hint: dont lie! 🙈)

2

how was your sleep last night? *

BEST NIGHT EVER! good eh...it was okay not so good didn't sleep a wink T_T

your morning mood? *

🥰 🙂 😐 😪 🙃 🙄

what are your goals for today?

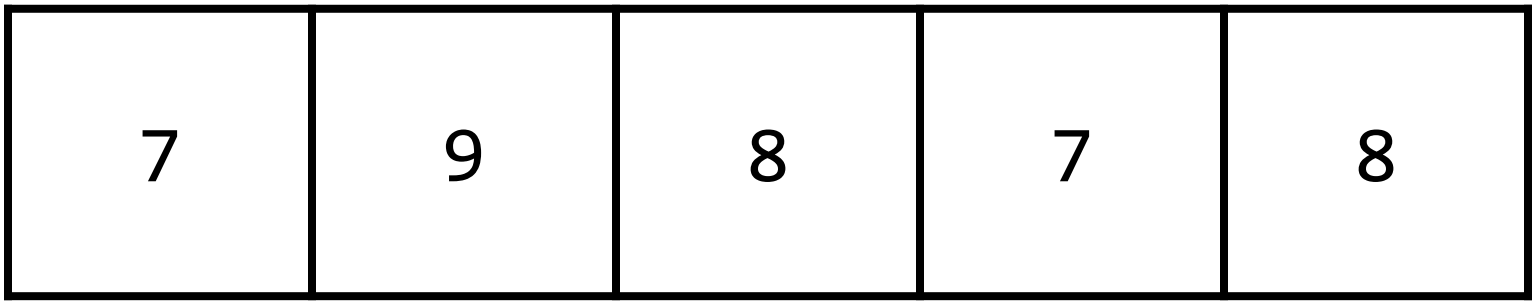
be brave, be bold, and be courageous!

did you have a dream last night? do you remember it?

record your dream last night if you remember :)

submit diary

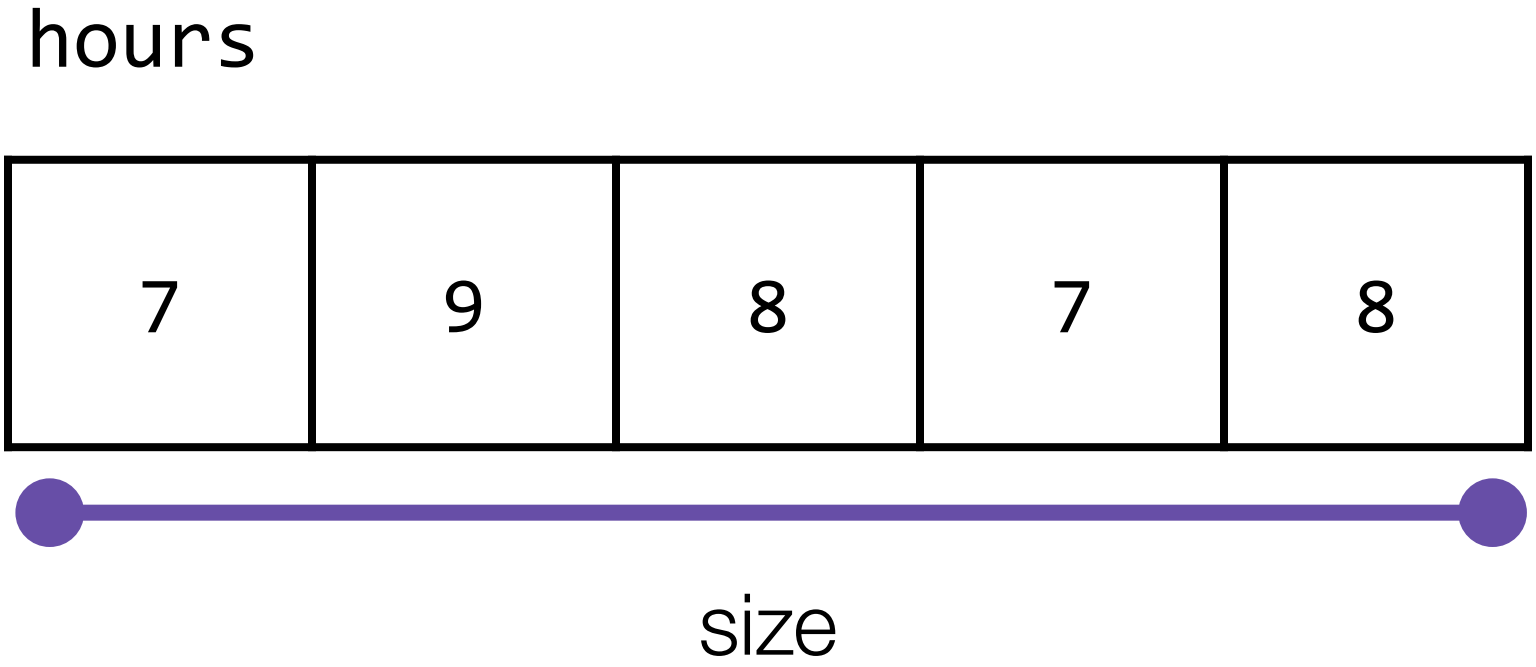
hours



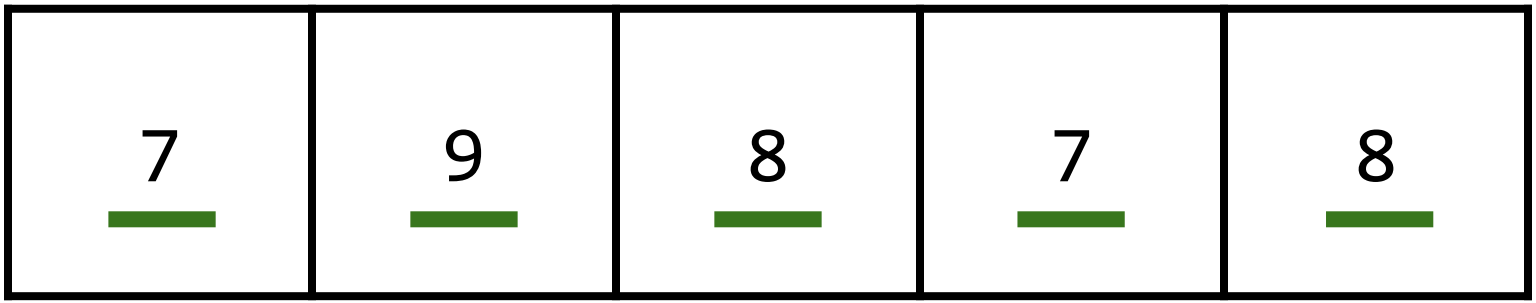
name

hours

7	9	8	7	8

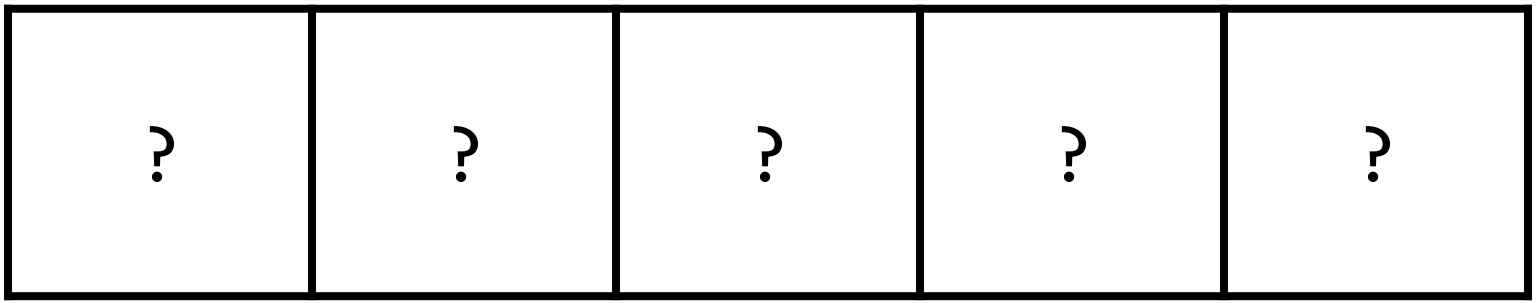

hours
7
9
8
7
8
size

`hours` type (int)



```
int hours[5];
```

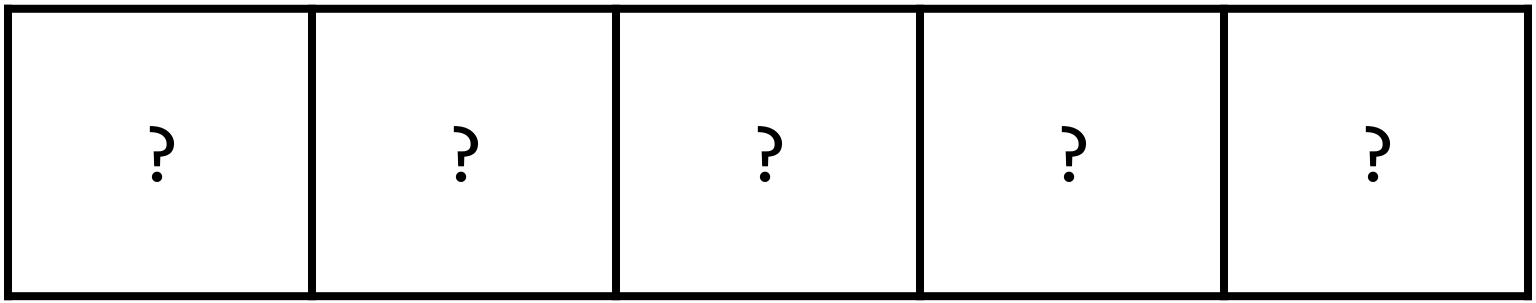
hours



name

```
int hours[5];
```

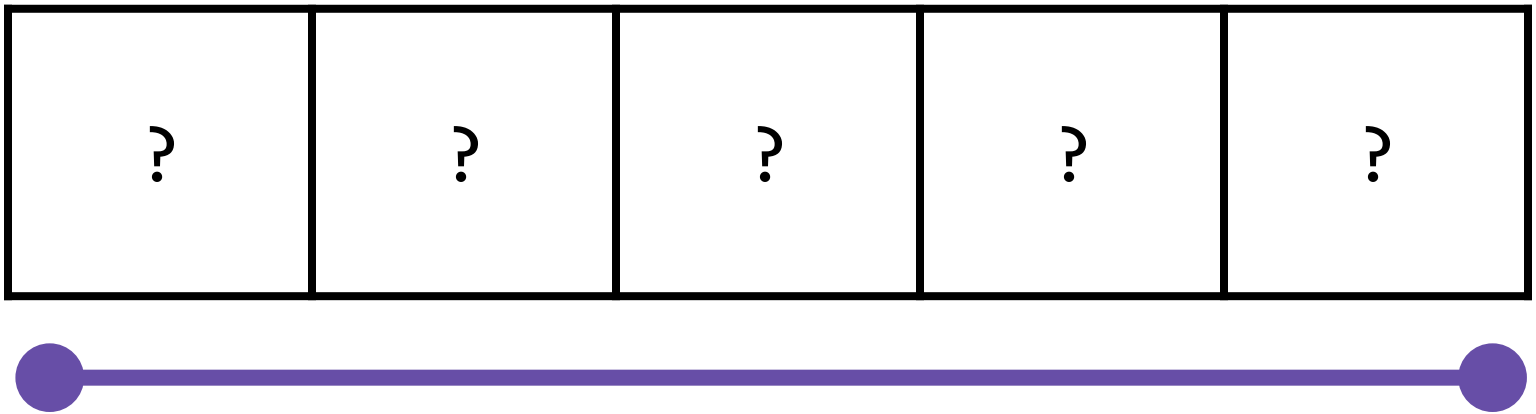
hours



size

```
int hours[5];
```

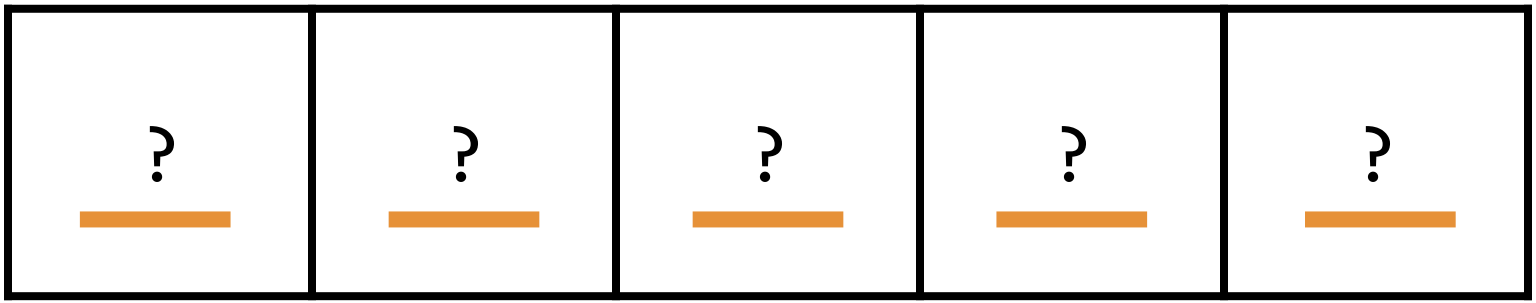
hours



type

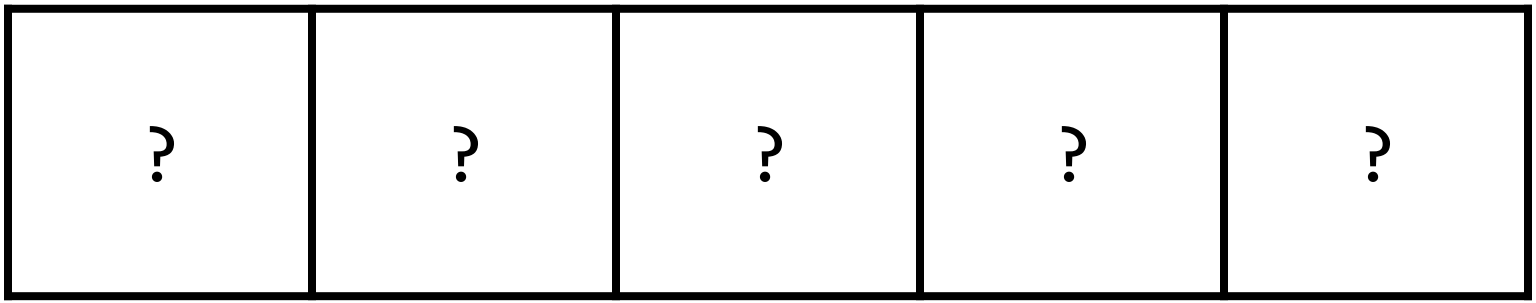
```
int hours[5];
```

hours



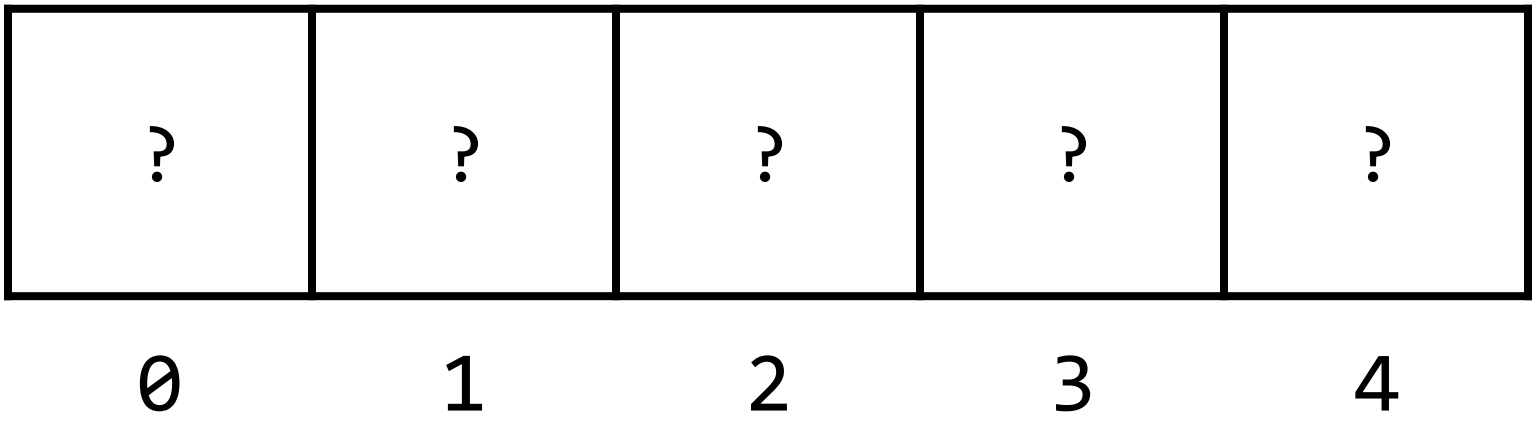
```
int hours[5];
```

hours



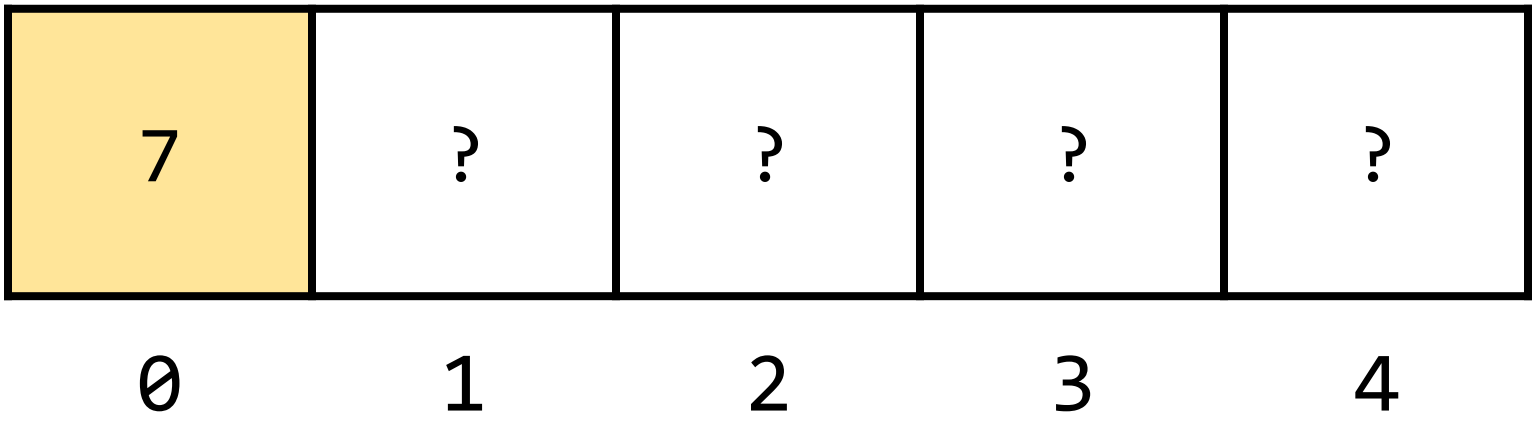
```
int hours[5];
```

hours



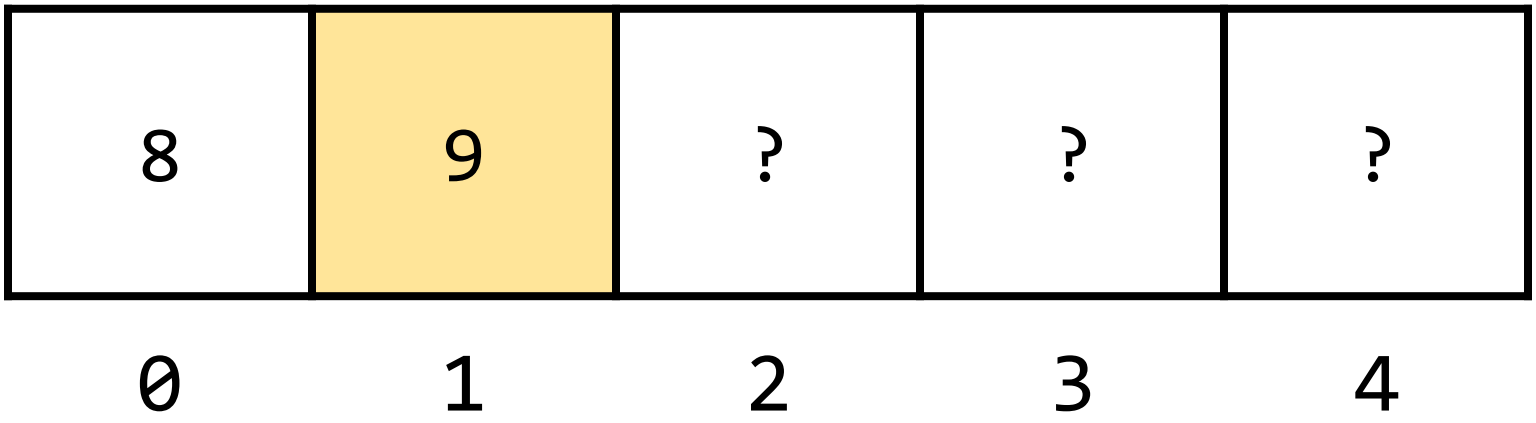
```
int hours[5];
hours[0] = 7;
```

hours



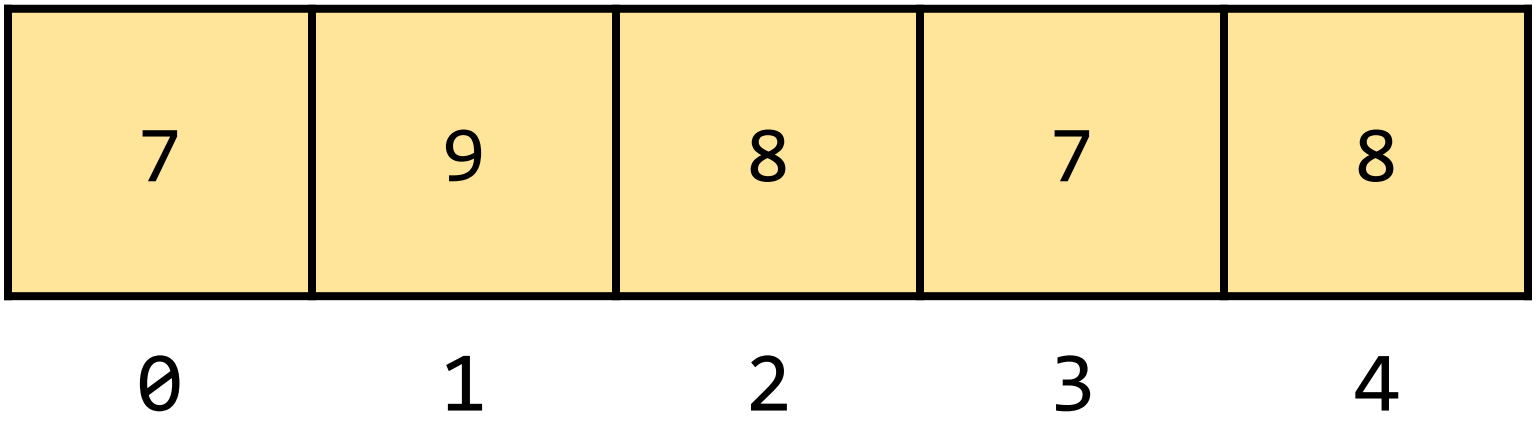
```
int hours[5];
hours[0] = 7;
hours[1] = 9;
```

hours



```
int hours[5] = {7, 9, 8, 7, 8};
```

hours



```
int hours[5] = {7, 9, 8, 7, 8};

for (int i = 0; i < 5; i++)
{
    printf("%i\n", hours[i]);
}
```

```c
int hours[5] = {7, 9, 8, 7, 8};

for (int i = 0; i < 5; i++)
{
    printf("%i\n", hours[i]);
}
```

Array Exercise

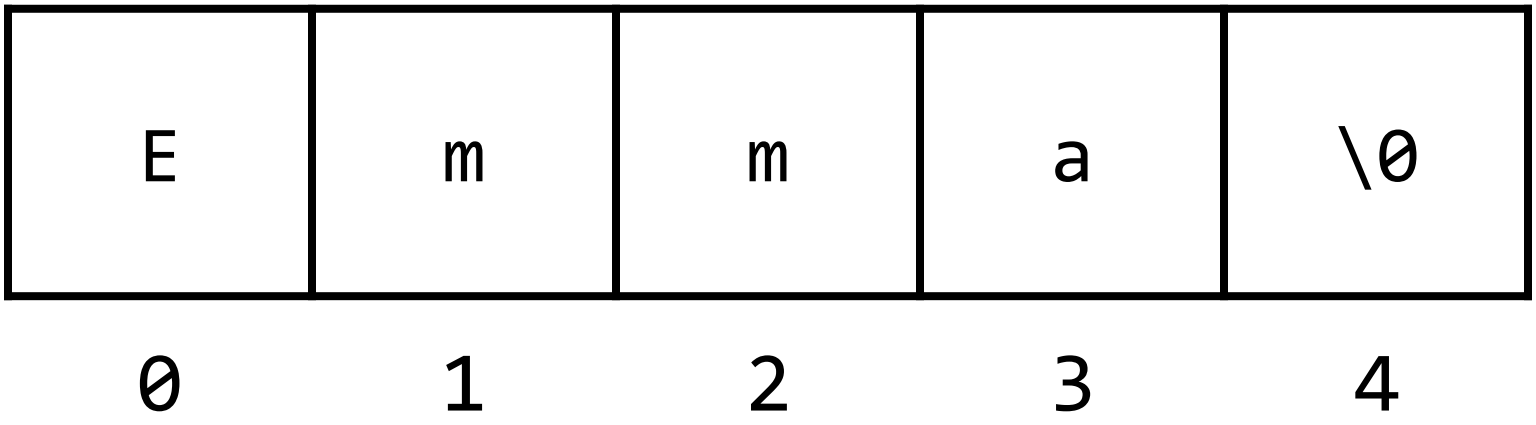
Create an array of size 5 where each element is two times the previous and the first element is 1.

Print the array, integer by integer.

Strings

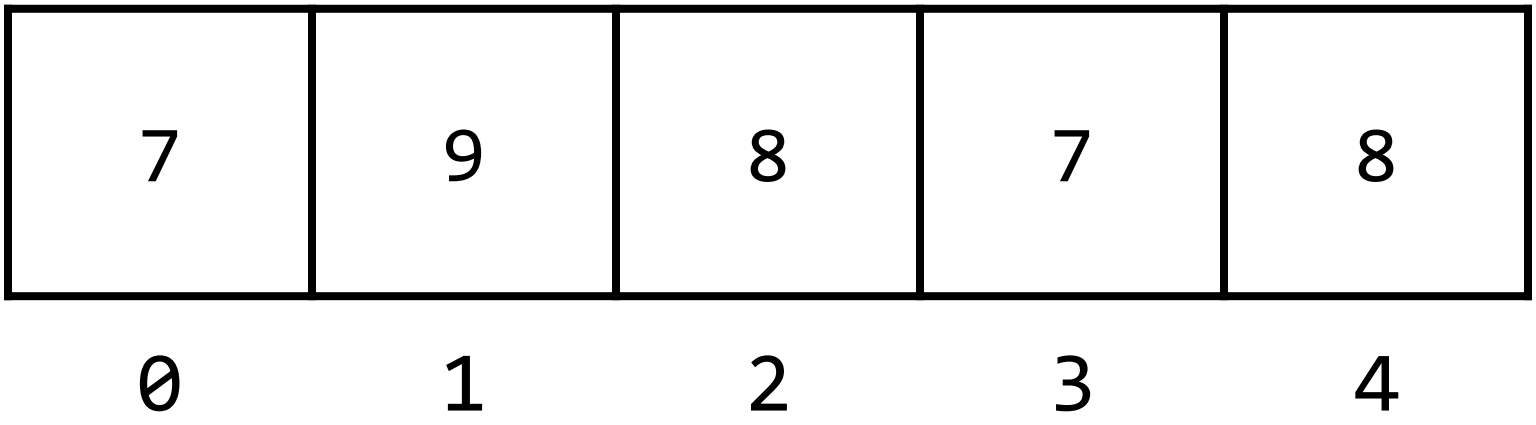
```
string name = "Emma";
```

name



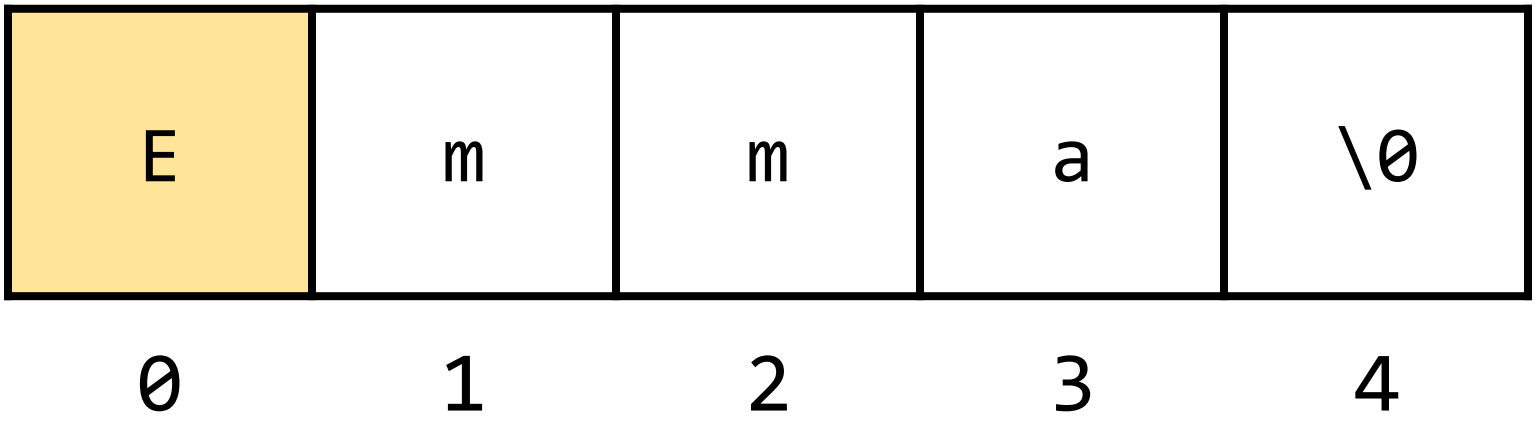
```
int hours[5] = {7, 9, 8, 7, 8};
```

hours



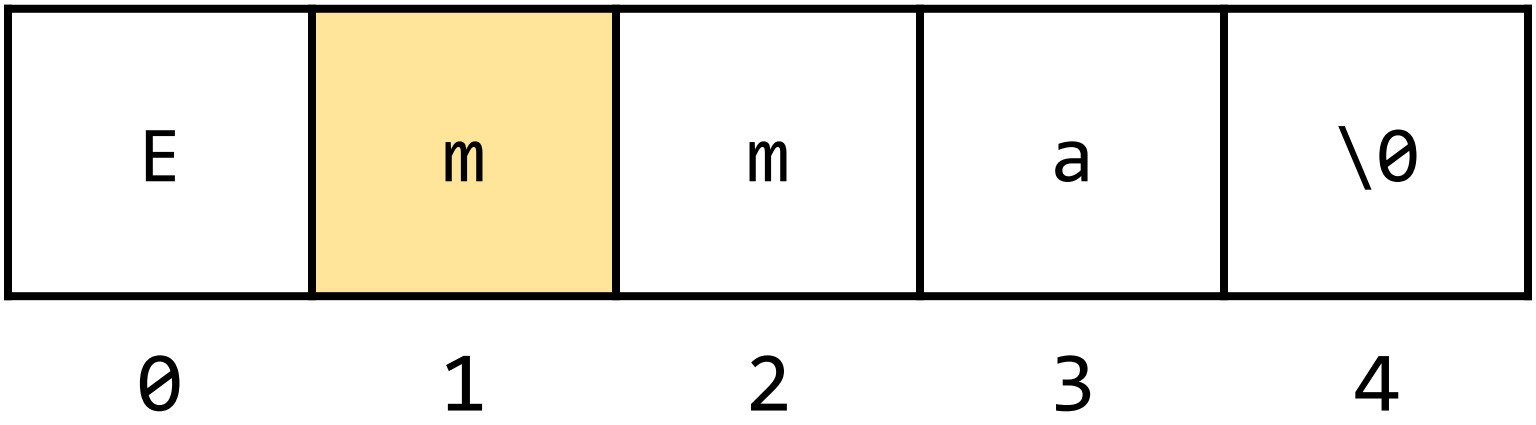
```
name[0];
```

```
name
```



```
name[1];
```

```
name
```



String Exercise

Create a string and print the string character by character.

```
string name = "Emma";
int length = strlen(name);

for (int i = 0; i < length; i++)
{
    printf("%c\n", name[i]);
}
```

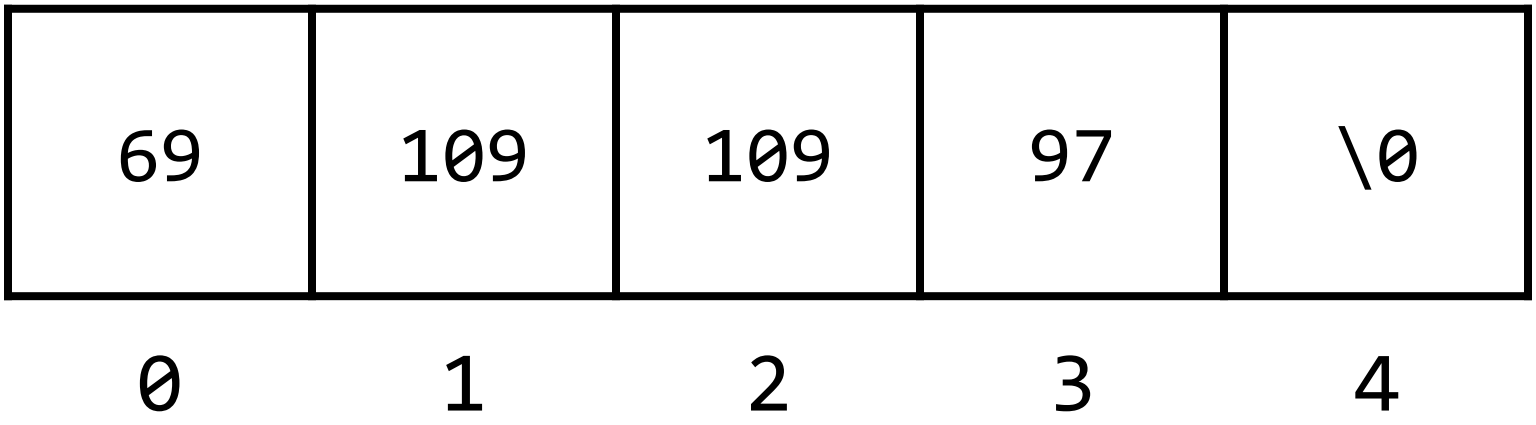
```
string name = "Emma";
int length = strlen(name);

for (int i = 0; i < length; i++)
{
    printf("%i\n", name[i]);
}
```

A	B	C	...	Z
65	66	67	...	90

a	b	c	...	z
97	98	99	...	122

```
string name = "Emma";
```

name



String Exercise II

Check if a lowercase string's characters are in alphabetical order. If yes, print "Yes". If no, print "No".

asciichart.com

Command-line Arguments

What are some examples of programs we've seen that take command-line arguments?

```
$ make mario
```

```
$ ./caesar 13
```

```
int calculate_quarters(int cents)
{
    ...
}
```

Function argument(s)

```
int calculate_quarters(int cents)
{
    ...
}
```

Return type

```
int calculate_quarters(int cents)
{
    ...
}
```

```
int main(void)
{
    ...
}
```

```
int main(int argc, string argv[])
{
    ...
}
```

```
$ make mario
```

argv[0] argv[1]

```
$ ./caesar 13
```

```
$ ./initials Carter Zenke
```

```
$ ./initials Carter Zenke
```

argv[1] argv[2]

```
$ ./initials Carter Zenke
```

argv[1][0] argv[2][0]

Lab

- Work an example yourself
- Write down exactly what you did
- Create a generalization (algorithm) after working multiple examples
- Test your algorithm by hand
- Translate your algorithm to code
- Find bugs in your code by running test cases
- Debug (and critique) your code

- Work an example yourself
- Write down exactly what you did
- Create a generalization (algorithm) after working multiple examples
- Test your algorithm by hand
- Translate your algorithm to code
- Find bugs in your code by running test cases
- Debug (and critique) your code

- What **syntax** should we use to access each individual character of a string?
- How should we get the **point value** of a character?
- How should our program handle **uppercase** and **lowercase** inputs differently?

- Work an example yourself
- Write down exactly what you did
- Create a generalization (algorithm) after working multiple examples
- Test your algorithm by hand
- Translate your algorithm to code
- Find bugs in your code by running test cases
- Debug (and critique) your code

If feeling more comfortable…

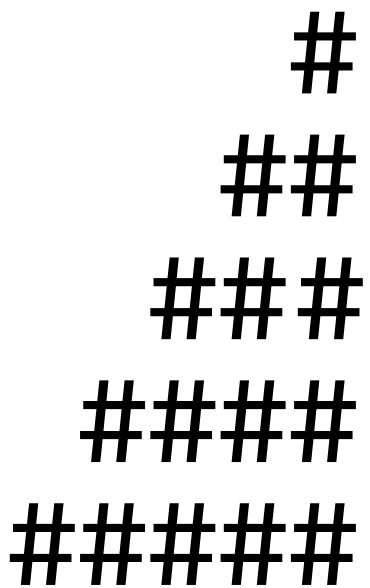
Our Scrabble program will accept any word, whether it's correctly spelled or not! How might you check to see if a user's input is part of a list of valid words?

Tutorials

Office Hours

cs50.ly/attend

cs50.ly/feedback



This was CS50.