This is CS50.

cs50.brianyu.me

Week 2

- Compiling
- Debugging
- Data Types
- Memory
- Arrays
- Strings
- Command-Line Arguments

What questions do you have?

Questions

- When to use command line arguments? argc and argv
-

Today

Arrays

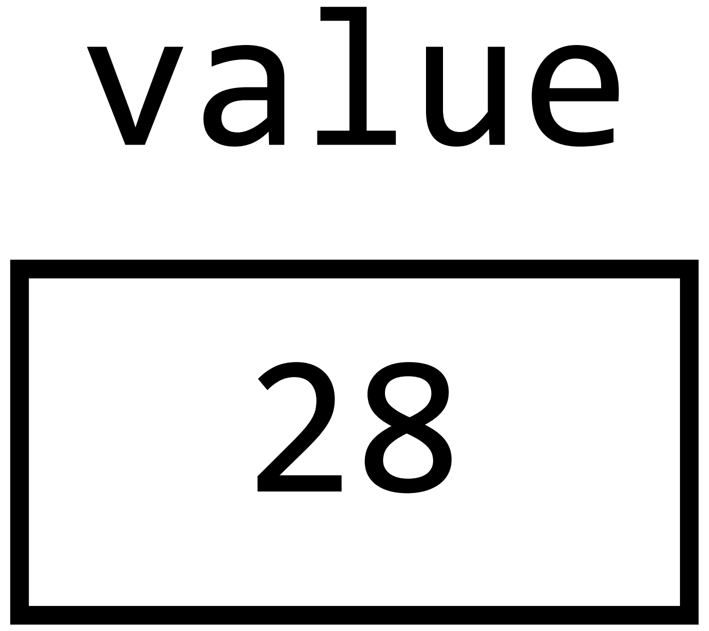
Strings

Command-Line Arguments

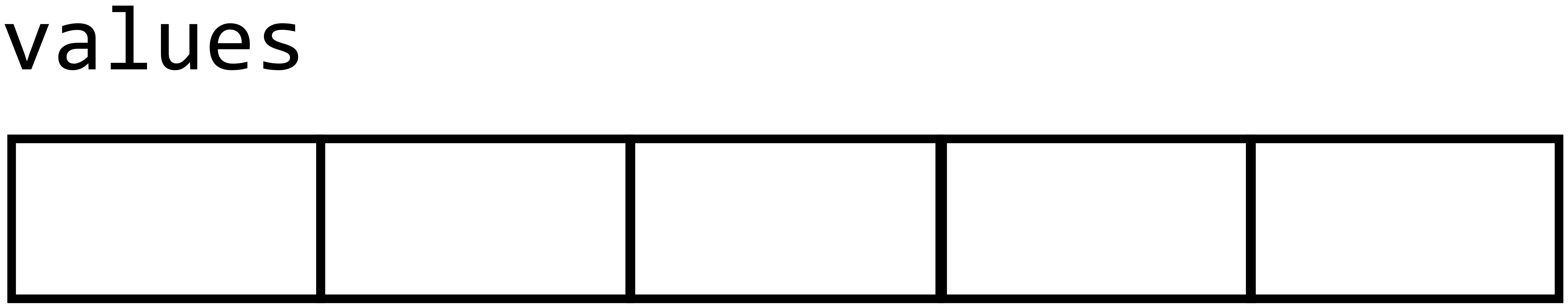
Part One

Arrays

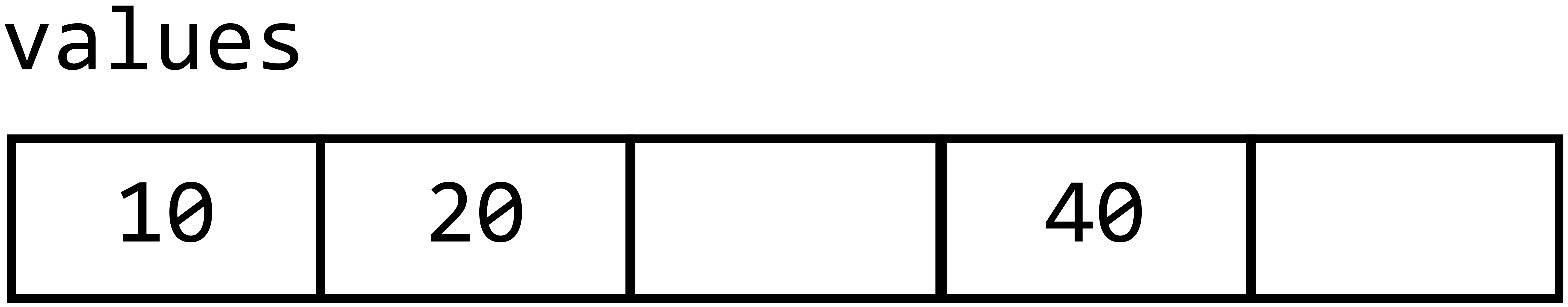
```
int value = 28;
```



```
int values[5];
```



```
int values[5];
values[0] = 10;
values[1] = 20;
values[3] = 40;
```



values

10	20		40	

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Work for

10

minutes

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Work for

9

minutes

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Work for

8

minutes

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Work for

7

minutes

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Work for

6

minutes

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Work for

5

minutes

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Work for

4

minutes

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Work for

3

minutes

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Work for

2

minutes

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Work for

1

minute

Exercise

Write a program that takes 5 integers and prints them in reverse order.

```
$ ./reverse
Number 1: 5
Number 2: 10
Number 3: 15
Number 4: 20
Number 5: 25
25 20 15 10 5
```

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

Work for

10

minutes

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

Work for

9

minutes

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

Work for

8

minutes

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

Work for

7

minutes

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

Work for

6

minutes

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

Work for

5

minutes

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

Work for

4

minutes

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

Work for

3

minutes

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

Work for

2

minutes

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

Work for

1

minute

Exercise

Write a program that takes 5 integers and prints out a bar chart of them.

```
$ ./chart
Number 1: 5
Number 2: 10
Number 3: 8
Number 4: 4
Number 5: 6
```

```
#####
##########
########
####
######
```

We'll continue in

10

minutes

We'll continue in

9

minutes

We’ll continue in

8

minutes

We'll continue in

7

minutes

We'll continue in

6

minutes

We'll continue in

5

minutes

We'll continue in

4

minutes

We’ll continue in

3

minutes

We'll continue in

2

minutes

We'll continue in

1

minute

Part Two
Strings

```
int main(void)
{
   printf("%c\n", 'A');
}
```

```
int main(void)
{
    printf("%i\n", 'A');
}
```

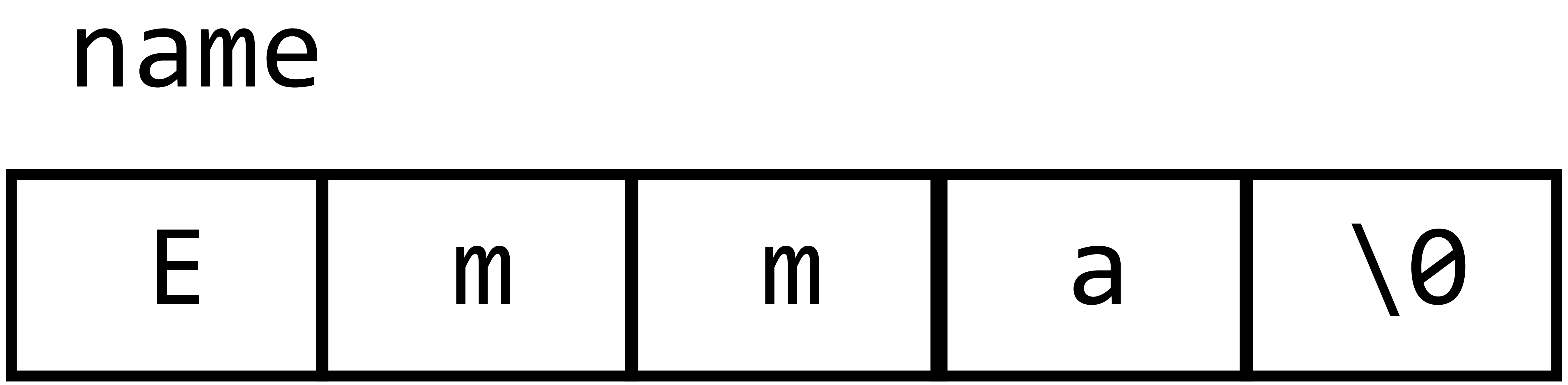
ASCII

A	B	C	D	E	...	Z
65	66	67	68	69	...	90

a	b	c	d	e	...	z
97	98	99	100	101	...	122

```
int main(void)
{
    printf("%i\n", 'A' + 1);
}
```

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



name

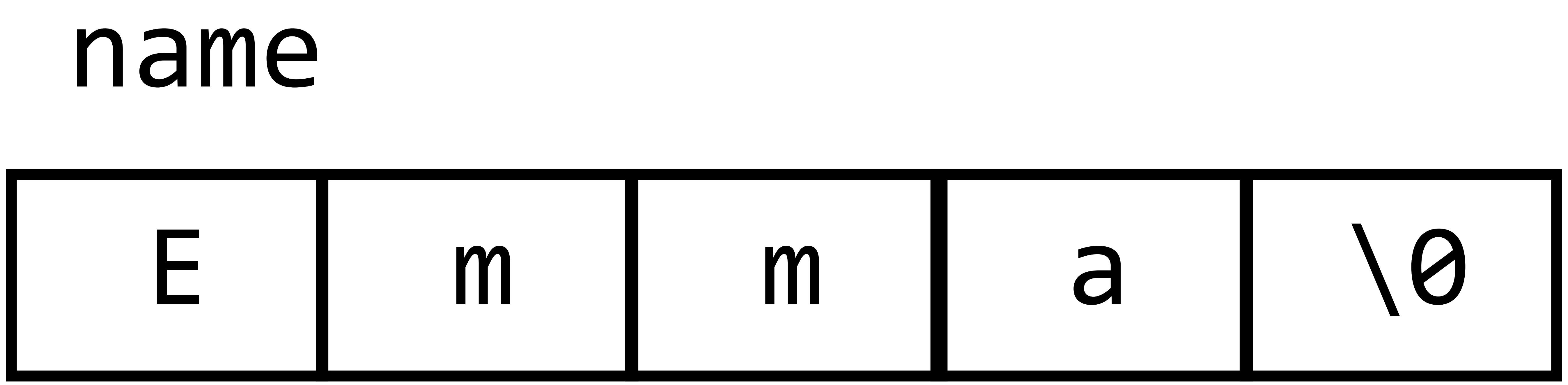
E	m	m	a	\0

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

name

69	109	109	97	\0

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



name

E	m	m	a	\0

strlen

Exercise

Update your reverse program to take a string as input, and print out the reverse of the string.

```
$ ./reverse
Text: Hello!
!olleH
```

Exercise

Update your reverse program to take a string as input, and print out the reverse of the string.

```
$ ./reverse
Text: Hello!
!olleH
```

Work for

7

minutes

Exercise

Update your reverse program to take a string as input, and print out the reverse of the string.

```
$ ./reverse
Text: Hello!
!olleH
```

Work for

6

minutes

Exercise

Update your reverse program to take a string as input, and print out the reverse of the string.

```
$ ./reverse
Text: Hello!
!olleH
```

Work for

5

minutes

Exercise

Update your reverse program to take a string as input, and print out the reverse of the string.

```
$ ./reverse
Text: Hello!
!olleH
```

Work for

4

minutes

Exercise

Update your reverse program to take a string as input, and print out the reverse of the string.

```
$ ./reverse
Text: Hello!
!olleH
```

Work for

3

minutes

Exercise

Update your reverse program to take a string as input, and print out the reverse of the string.

```
$ ./reverse
Text: Hello!
!olleH
```

Work for

2

minutes

Exercise

Update your reverse program to take a string as input, and print out the reverse of the string.

```
$ ./reverse
Text: Hello!
!olleH
```

Work for

1

minute

Exercise

Update your reverse program to take a string as input, and print out the reverse of the string.

```
$ ./reverse
Text: Hello!
!olleH
```

Exercise

Write a program **`palindrome.c`** that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Exercise

Write a program **`palindrome.c`** that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Work for

10

minutes

Exercise

Write a program **`palindrome.c`** that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Work for

9

minutes

Exercise

Write a program **`palindrome.c`** that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Work for

8

minutes

Exercise

Write a program **`palindrome.c`** that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Work for

7

minutes

Exercise

Write a program **`palindrome.c`** that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Work for
6
minutes

Exercise

Write a program **`palindrome.c`** that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Work for

5

minutes

Exercise

Write a program **`palindrome.c`** that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Work for

4

minutes

Exercise

Write a program `palindrome.c` that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Work for

3

minutes

Exercise

Write a program **`palindrome.c`** that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Work for

2

minutes

Exercise

Write a program **palindrome.c** that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Work for
1
minute

Exercise

Write a program **`palindrome.c`** that takes a string as input, and determines whether it is a palindrome (the same backwards and forwards).

```
$ ./palindrome
Text: racecar
PALINDROME
$ ./palindrome
Text: jellyfish
NOT PALINDROME
```

Exercise

Write a program **`consonants.c`** that takes a string as input, and prints it out without vowels.

```
$ ./consonants
Text: This is CS50.
Ths s CS50.
```

Exercise

Write a program **consonants.c** that takes a string as input, and prints it out without vowels.

```
$ ./consonants
Text: This is CS50.
Ths s CS50.
```

Work for

7

minutes

Exercise

Write a program **`consonants.c`** that takes a string as input, and prints it out without vowels.

```
$ ./consonants
Text: This is CS50.
Ths s CS50.
```

Work for

6

minutes

Exercise

Write a program **`consonants.c`** that takes a string as input, and prints it out without vowels.

```
$ ./consonants
Text: This is CS50.
Ths s CS50.
```

Work for

5

minutes

Exercise

Write a program **consonants.c** that takes a string as input, and prints it out without vowels.

```
$ ./consonants
Text: This is CS50.
Ths s CS50.
```

Work for

4

minutes

Exercise

Write a program **`consonants.c`** that takes a string as input, and prints it out without vowels.

```
$ ./consonants
Text: This is CS50.
Ths s CS50.
```

Work for

3

minutes

Exercise

Write a program **consonants.c** that takes a string as input, and prints it out without vowels.

```
$ ./consonants
Text: This is CS50.
Ths s CS50.
```

Work for

2

minutes

Exercise

Write a program **`consonants.c`** that takes a string as input, and prints it out without vowels.

```
$ ./consonants
Text: This is CS50.
Ths s CS50.
```

Work for

1

minute

Exercise

Write a program **`consonants.c`** that takes a string as input, and prints it out without vowels.

```
$ ./consonants
Text: This is CS50.
Ths s CS50.
```

We'll continue in

10

minutes

We'll continue in

9

minutes

We'll continue in

8

minutes

We'll continue in

7

minutes

We'll continue in

6

minutes

We'll continue in

5

minutes

We'll continue in

4

minutes

We’ll continue in

3

minutes

We'll continue in

2

minutes

We'll continue in

1

minute

Part Three

Command-Line Arguments

```
$ ./cash
```

```
$ make mario
```

```
$ clang -o hello hello.c
$ ./cash

$ make mario

$ clang -o hello hello.c
$ ./cash
   argv[0]

$ make mario
  argv[0] argv[1]

$ clang -o hello hello.c
  argv[0] argv[1] argv[2] argv[3]
```

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

Argument Count   Argument Vector

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

Exercise

Write a program **`capitalize.c`** that capitalizes a name provided as command-line arguments.

```
$ ./capitalize rodrigo daboin sanchez

Rodrigo Daboin Sanchez
```

Modulo

Modulo

- `a % b` returns the remainder when `a` is divided by `b`

Exercise

Write a program **`leapyear.c`** that tells you if a year is a leap year.

```
$ ./leapyear 2019
Not a leap year
$ ./leapyear 2020
Leap year
```

Problem Set 2

Problem Set 2

- Readability
- One of:
    - Caesar
    - Substitution

This is CS50.