RANDOM NUMBER GENERATOR

```
#include <stdlib.h> /* rand() */

// …

int number = rand() % 10 + 1;
```

Generates a random number between 1 and 10.

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RANDOM NUMBER GENERATOR

```
#include <stdlib.h> /* rand() */

// …

int number = rand() % 10 + 1;
```

Generates a random number between 1 and 10.

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RANDOM NUMBER GENERATOR

```
#incl                     () */

// …

int n                     1;
```

```
1
1
1
2
5
5
5
```

erates a random number between 1 and 10.

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RANDOM NUMBER GENERATOR

```
#incl
// …
int n
```

```
1
1
1
2
5
5
5
```

```
1
1
1
2
5
5
5
```

s a random number
veen 1 and 10.

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RANDOM NUMBER GENERATOR

```
#incl

// …

int n
```

```
1
1
1
2
5
5
5
```

```
1
1
1
2
5
5
5
```

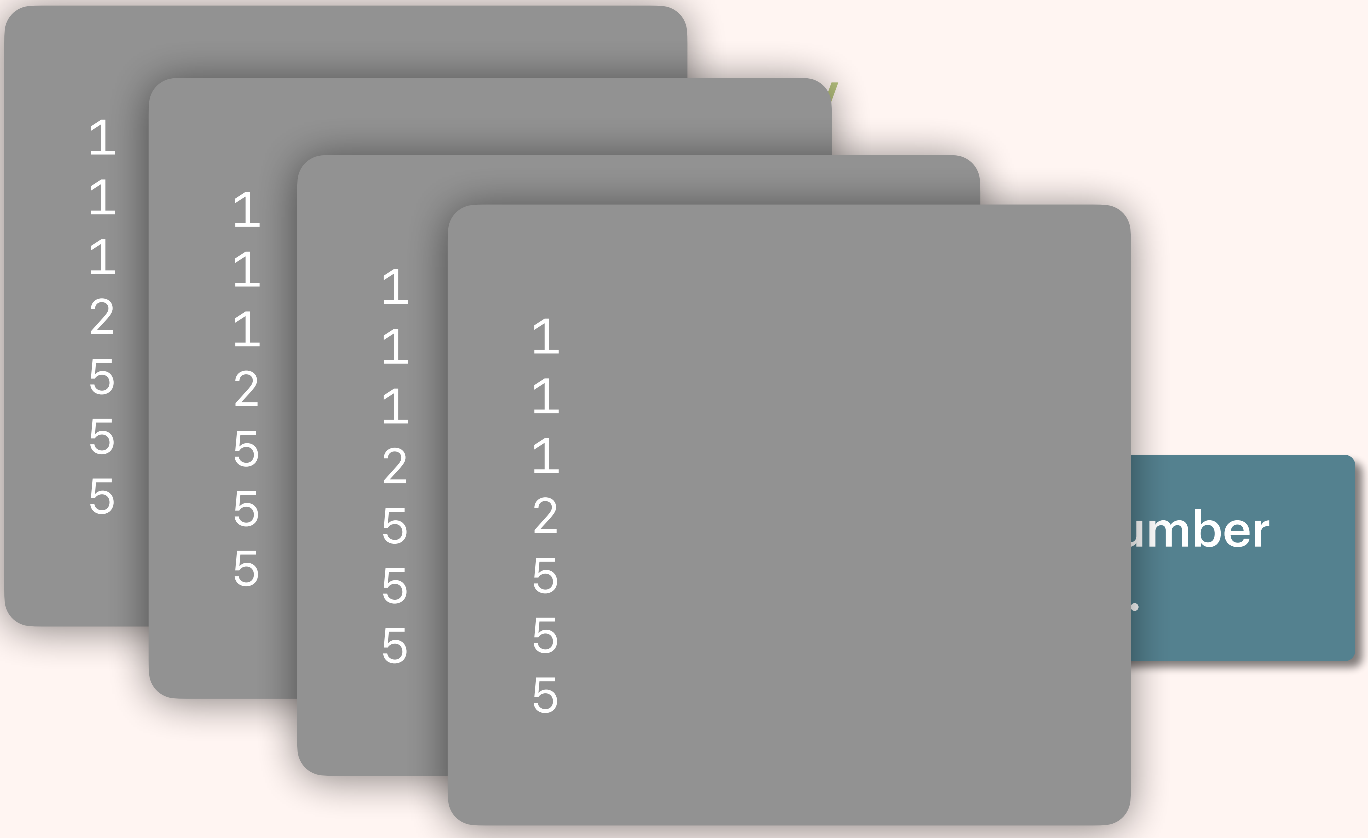
```
1
1
1
2
5
5
5
```

dom number
and 10.

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RANDOM NUMBER GENERATOR

```
#incl

// …

int n
```



```
1
1
1
2
5
5
5
```

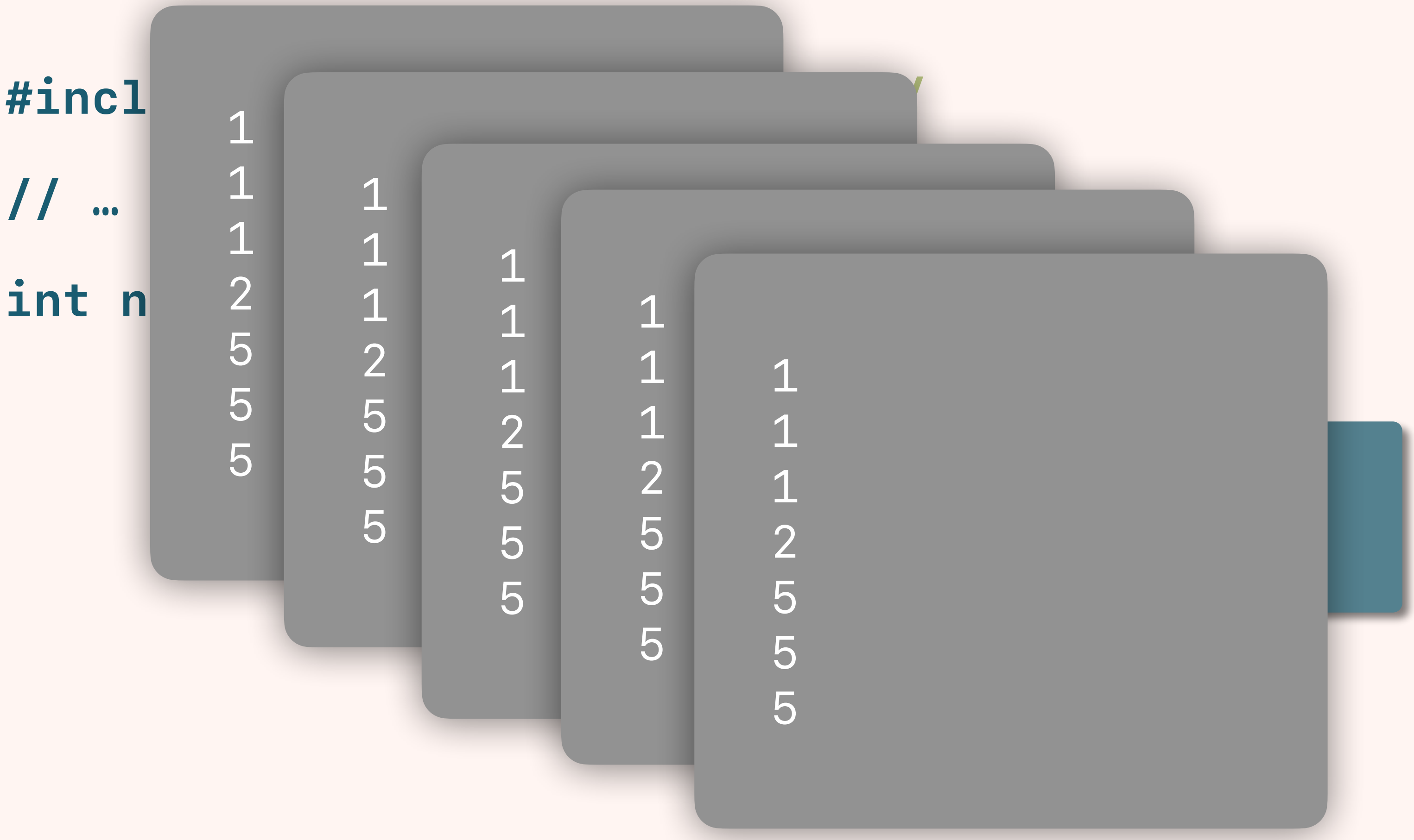
```
1
1
1
2
5
5
5
```

```
1
1
1
2
5
5
5
```

```
1
1
1
2
5
5
5
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RANDOM NUMBER GENERATOR


```
#incl
// …
int n
```

```
1
1
1
2
5
5
5
```

```
1
1
1
2
5
5
5
```

```
1
1
1
2
5
5
5
```

```
1
1
1
2
5
5
5
```

```
1
1
1
2
5
5
5
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

PA4 stuff!!!

RANDOM NUMBER GENERATOR

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

PA4 stuff!!!

RANDOM NUMBER GENERATOR

```
#incl
// …
int n
```

```
1
1
1
2
5
5
5
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

PA4 stuff!!!

COMPUTERS CAN’T BE RANDOM

* massive simplification

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

COMPUTERS CAN'T BE RANDOM

Think Deck of Cards

* massive simplification

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

COMPUTERS CAN'T BE RANDOM

Think Deck of Cards

(but with 4,294,967,296 cards)

* massive simplification

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

COMPUTERS CAN'T BE RANDOM

Think Deck of Cards

(but with 4,294,967,296 cards)

* massive simplification

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

COMPUTERS CAN'T BE RANDOM

Think Deck of Cards
(but with 4,294,967,296 cards)

Calling `rand()` is taking the top card.

* massive simplification

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

COMPUTERS CAN'T BE RANDOM

Think Deck of Cards
(but with 4,294,967,296 cards)

Calling `rand()` is taking the top card.

We need to tell `rand()` how to shuffle the deck.

* massive simplification

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

COMPUTERS CAN'T BE RANDOM

Think Deck of Cards
(but with 4,294,967,296 cards)

Calling `rand()` is taking the top card.

We need to tell `rand()` how to shuffle the deck.

This is called "seeding" or "seed"

* massive simplification

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

PA4 stuff!!!

COMPUTERS CAN'T BE RANDOM

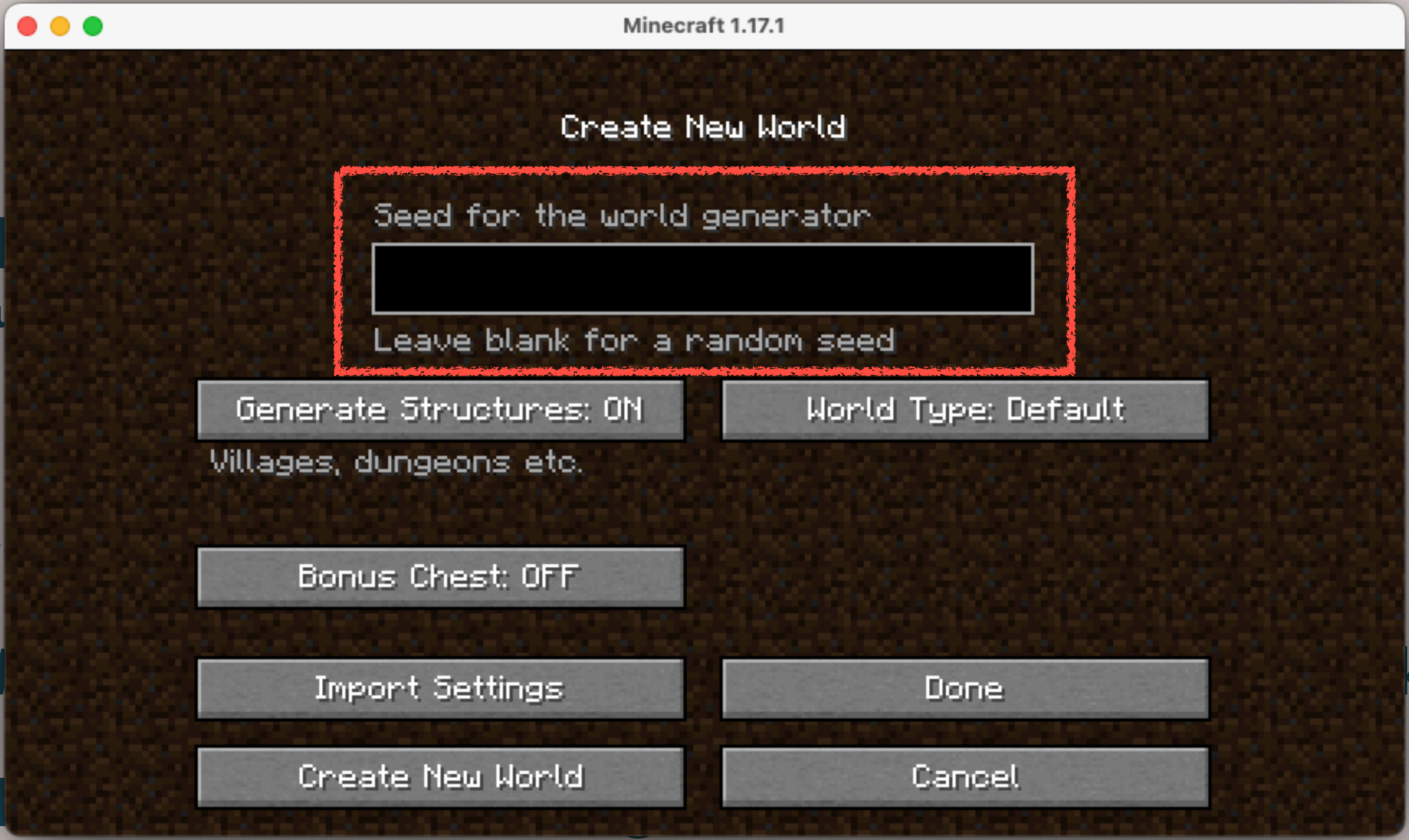
TI

(bu

C

W k.

TI



Create New World

Seed for the world generator

Leave blank for a random seed

Generate Structures: ON

Villages, dungeons etc.

World Type: Default

Bonus Chest: OFF

* massive simplification

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RNG & TIME

For CPT_S 121, use the current time as the seed.

`time()` **returns the** number of seconds since 1/1/1970 (GMT).

Some history: this is called the UNIX EPOCH

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RNG & TIME

For CPT_S 121, use the current time as the seed.

`time()` **returns the** number of seconds since 1/1/1970 (GMT).

Some history: this is called the UNIX EPOCH

September 30, 2021 1:40:22 PM GMT-07:00 (PST)

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RNG & TIME

For CPT_S 121, use the current time as the seed.

`time()` **returns the** number of seconds since 1/1/1970 (GMT).

Some history: this is called the UNIX EPOCH

September 30, 2021 1:40:22 PM GMT-07:00 (PST)

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RNG & TIME

For CPT_S 121, use the current time as the seed.

`time()` **returns the** number of seconds since 1/1/1970 (GMT).

Some history: this is called the UNIX EPOCH

September 30, 2021 1:40:22 PM GMT-07:00 (PST)

$\downarrow$

September 30, 2021 8:40:22 PM GMT

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RNG & TIME

For CPT_S 121, use the current time as the seed.

`time()` **returns the** number of seconds since 1/1/1970 (GMT).

Some history: this is called the UNIX EPOCH

September 30, 2021 1:40:22 PM GMT-07:00 (PST)

September 30, 2021 8:40:22 PM GMT

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RNG & TIME

For CPT_S 121, use the current time as the seed.

`time()` **returns the** number of seconds since 1/1/1970 (GMT).

Some history: this is called the UNIX EPOCH

$$\begin{tikzcd}
\text{September 30, 2021 1:40:22 PM GMT-07:00 (PST)} \arrow[d, bend right] \\
\text{September 30, 2021 8:40:22 PM GMT} \arrow[d, bend right] \\
\text{1633034422}
\end{tikzcd}$$

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

RNG & TIME

For CPT_S 121, use the current time as the seed.

`time()` **returns the** number of seconds since 1/1/1970 (GMT).

Some history: this is called the UNIX EPOCH

September 30, 2021 1:40:22 PM GMT-07:00 (PST)

September 30, 2021 8:40:22 PM GMT

1633034422

Just like how char is just a character as int, the epoch is just a time as int.

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

TIME

```
#include <stdlib.h>    /* rand() */
#include <time.h>      /* time() */


srand(time(NULL));     // Call this only ONCE in your program

// …

int number = rand() % 10 + 1;
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

TIME

```
#include <stdlib.h>    /* rand() */
#include <time.h>      /* time() */


srand(time(NULL));     // Call this only ONCE in your program

// …

int number = rand() % 10 + 1;
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

PA4 stuff!!!

TIME

```
#incl                                    nd() */
#incl                                    me() */


srand                                    l this only ONCE in your program

// …

int n                                    1;
```

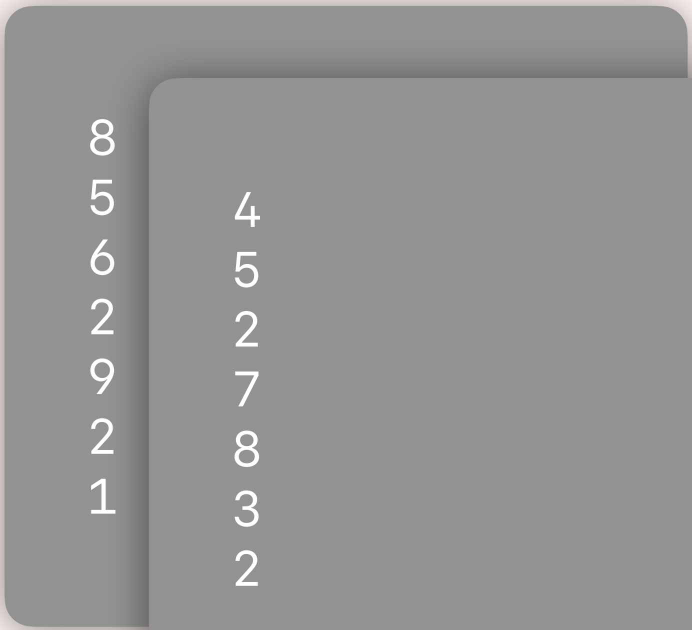
```
8
5
6
2
9
2
1
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

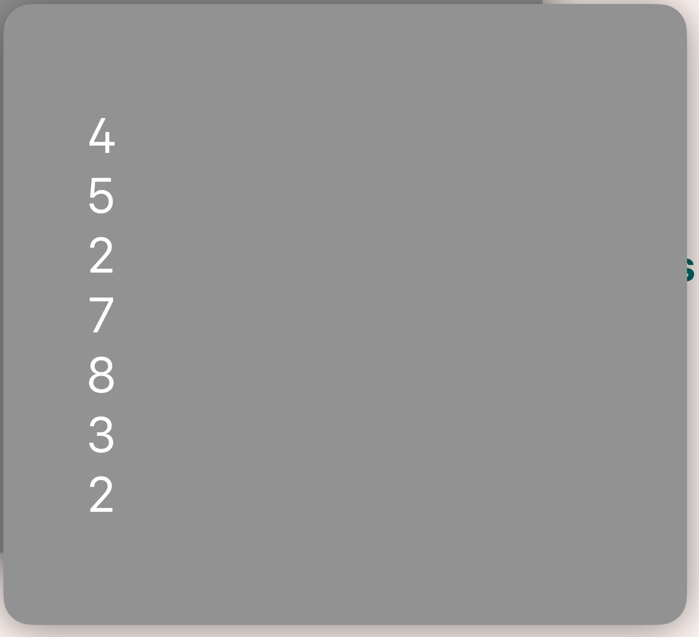
PA4 stuff!!!

TIME

```
#incl                                   */
#incl                                   */

srand                                 s only ONCE in your program

// …

int n
```



```
8
5
6
2
9
2
1
```



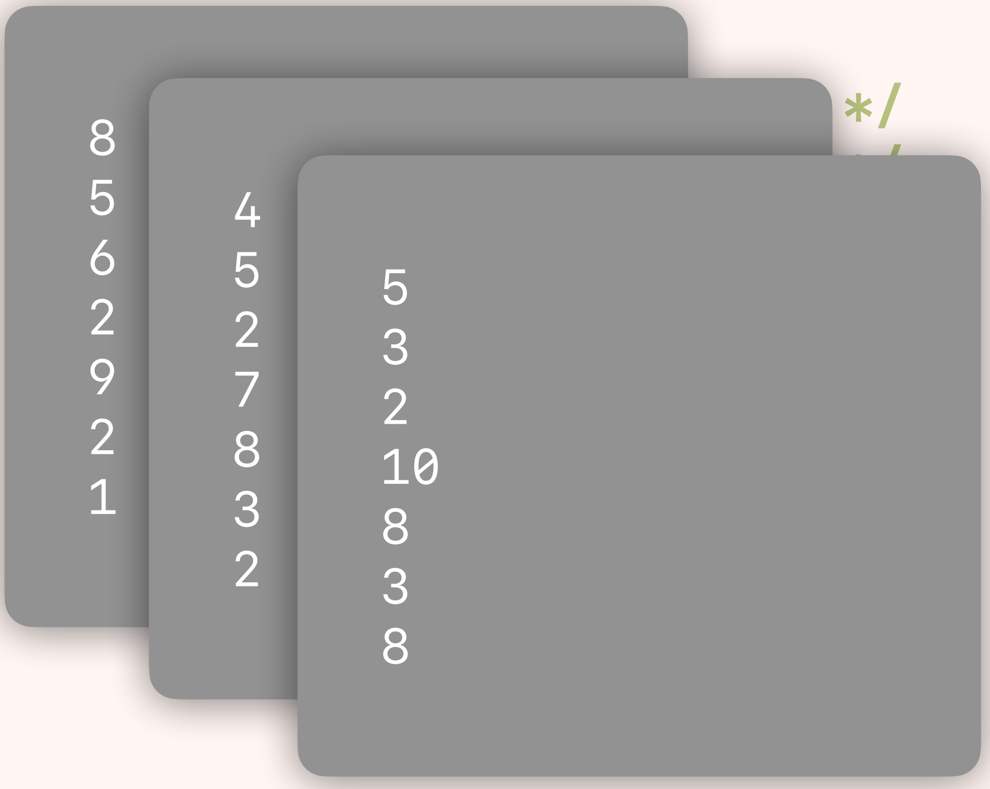
```
4
5
2
7
8
3
2
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

PA4 stuff!!!

TIME

```
#incl                              */
#incl

srand                                    ONCE in your program

// …

int n
```



```
8
5
6
2
9
2
1
```

```
4
5
2
7
8
3
2
```

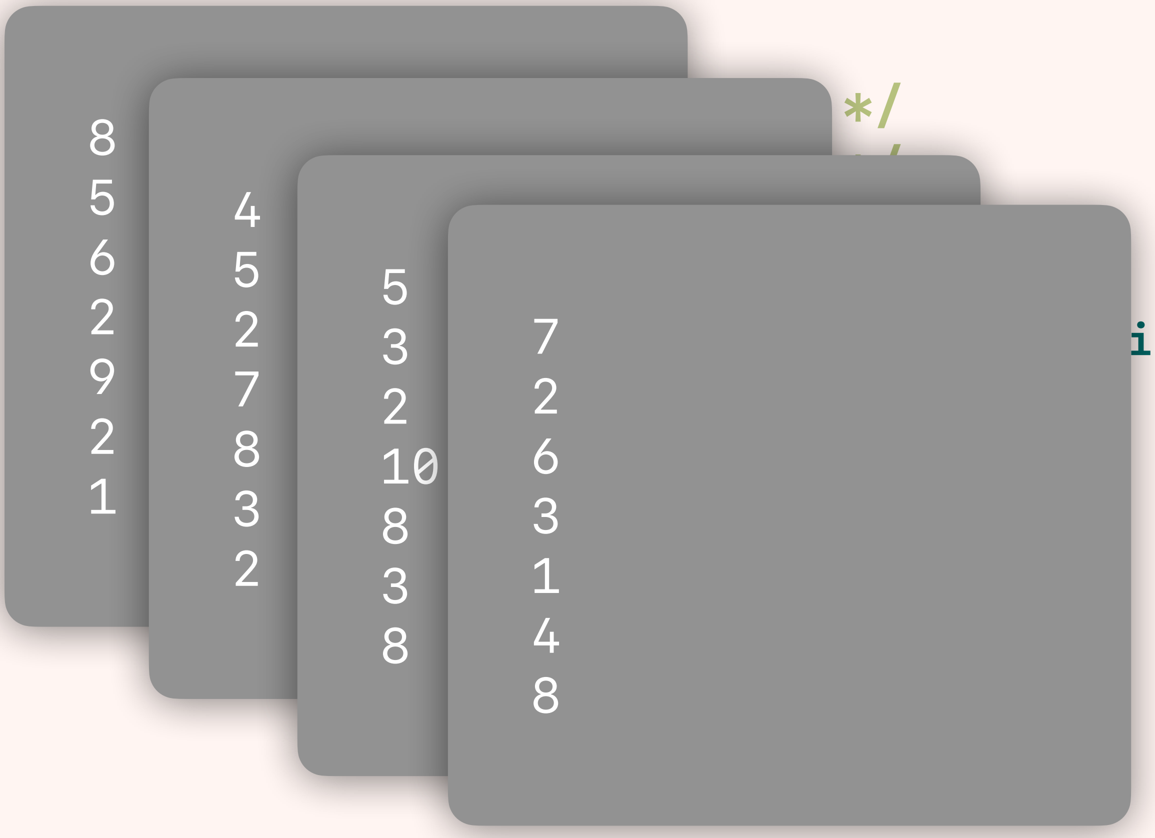
```
5
3
2
10
8
3
8
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

PA4 stuff!!!

TIME

```
#incl                                  */
#incl

srand                                          in your program

// …

int n
```



```
8
5
6
2
9
2
1
```

```
4
5
2
7
8
3
2
```

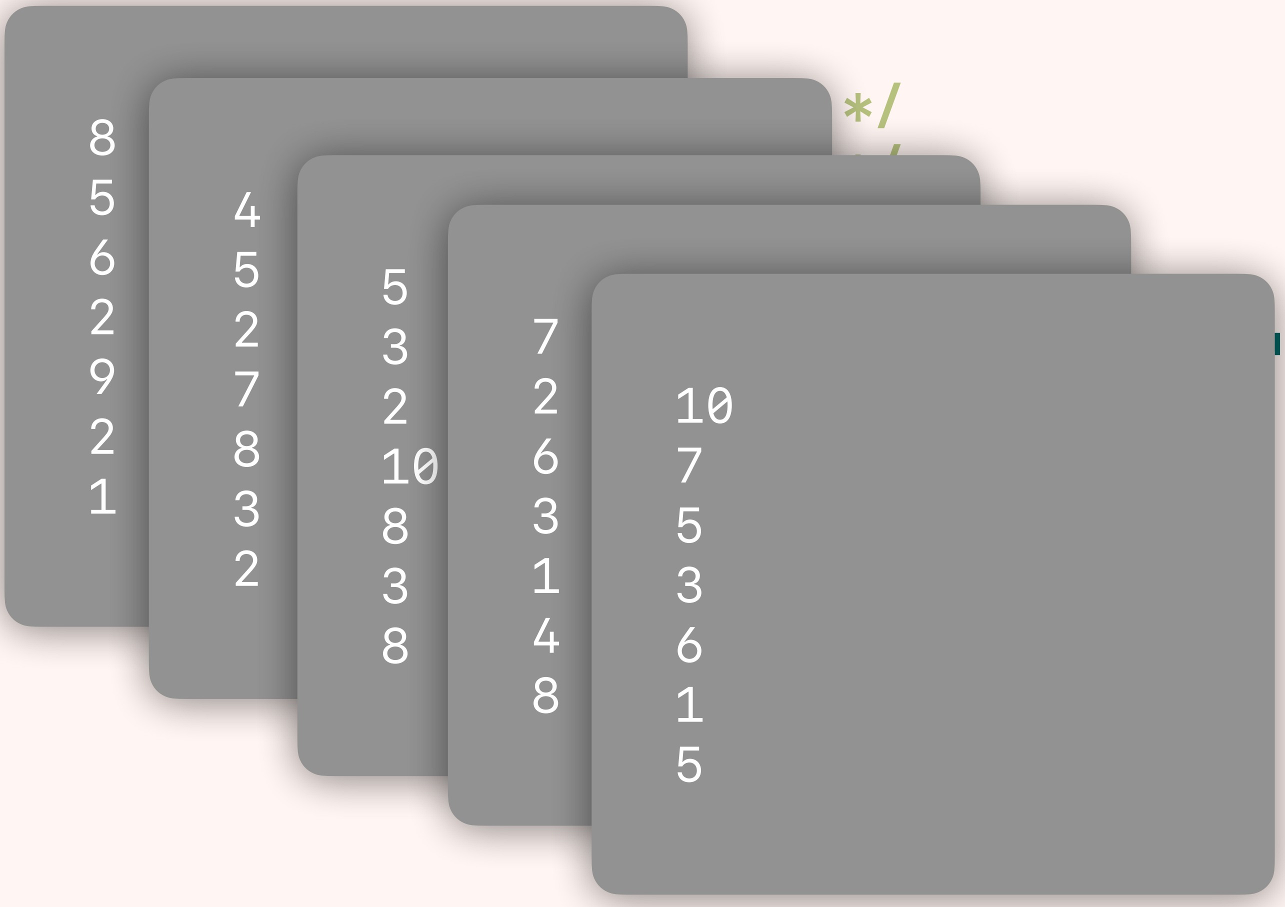
```
5
3
2
10
8
3
8
```

```
7
2
6
3
1
4
8
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

TIME

```
#incl                               */
#incl

srand                                                  ur program

// …

int n
```



```
8
5
6
2
9
2
1
```

```
4
5
2
7
8
3
2
```

```
5
3
2
10
8
3
8
```

```
7
2
6
3
1
4
8
```

```
10
7
5
3
6
1
5
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

PA4 stuff!!!

TIME

```
#incl                 */
#incl

srand                 ur program

// …

int n
```

```
8
5
6
2
9
2
1
```

```
4
5
2
7
8
3
2
```

```
5
3
2
10
8
3
8
```

```
7
2
6
3
1
4
8
```

```
10
7
5
3
6
1
5
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

PA4 stuff!!!

TIME

```
#incl                                  */
#incl

srand                                  ur program

// …

int n
```

```
8
5
6
2
9
2
1
```

```
4
5
2
7
8
3
2
```

```
5
3
2
10
8
3
8
```

```
7
2
6
3
1
4
8
```

```
10
7
5
3
6
1
5
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

PA4 hint

```c
#include <stdlib.h> /* rand() */
#include <time.h>   /* time() */

// Generates a random integer between the given bounds, inclusive.
int randomNumber(int min, int max) {
    return rand() % max + min;
}

int main(void) {
    srand(time(NULL));                  // Seed generator with current time.

    int n1 = randomNumber(1, 10);    // Get a random number.
    int n2 = randomNumber(1, 10);    // Get another random number.

    // ... blah blah blah

    return 0;
}
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

Commonly missed questions for Exam 1

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

ASCII

American Standard Code for Information Interchange

Character encoding, represents English characters as integers.

- **Developed in the 1960s.**
- **Unicode is the modern standard, and is primarily used today.**
- **ASCII uses one byte (8 bits) to represent one character.**
- **Unicode uses any number of 8 or 16 bits to represent one "character" or emoji 😎😝🤩🙀🧠🐔🐥🥬🥦🏳️‍🌈🇺🇸.**

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

WRITE A FUNCTION…

~~`int main(void) { … }`~~

```
int read_int(FILE *inFile) { … }
```

```
double quadratic_formula(double a, double b, double c) { … }

int is_palindrome(char c1, char c2, char c3, char c4) { … }
```

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License

FUNCTION ARGUMENTS ARE VARIABLES

```
double quadratic_formula(double a, double b, double c) {
    ~~double a, b, c;~~
    ~~printf("Please enter value for a: ");~~
    ~~scanf("%lf", &a);~~

    double x = ...
    return x;
}
```

a, b, c are already variables since they are declared as function arguments

your functions are not responsible for asking the user to input data.

Author: Alan Chu (ualch9@gmail.com)
https://github.com/ualch9/cpts121-lab-presentations
Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License