

CS 103000

Prof. Madeline Blount

Week 7:

VECTORS (cont.)

Attendance:

<https://cs103-proton2.glitch.me/>



Dall-E 2: cats learning C++ in the forest on '90's technology

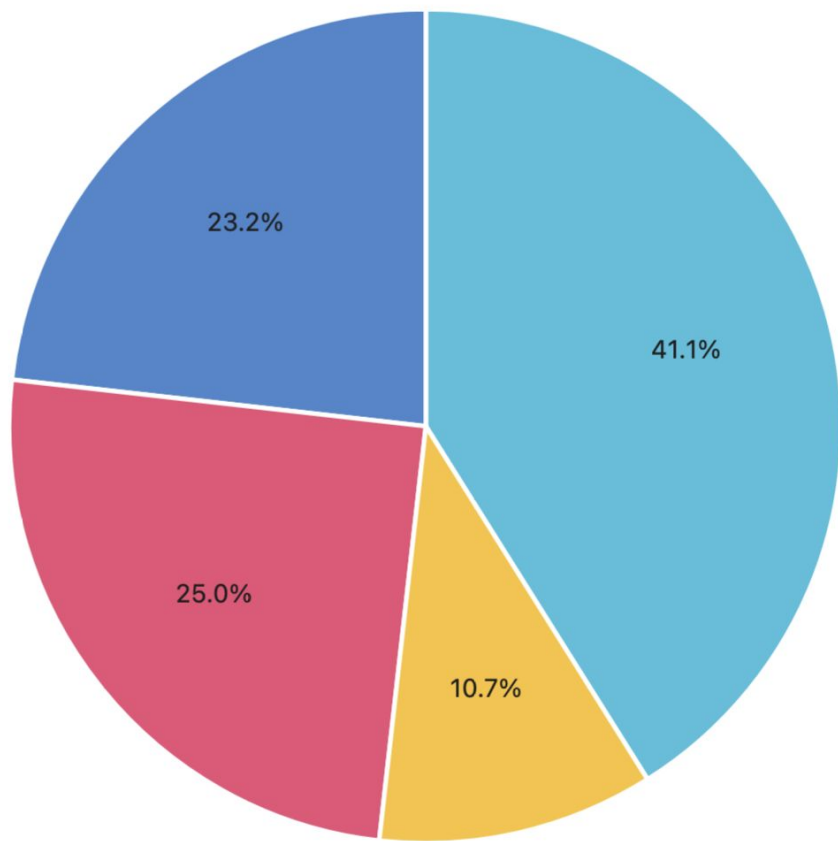


## MIDTERM!

- Review on 10.21, based on survey
- HACKATHON TEAMS also on 10.21, attendance required
- You can use any resources (with citation!)
- In-class work Wed. 10.23, Fri. 10.25
- Code due + individual questions due 10.25

### SUGGESTED REVIEW:

- In-class code for Project Euler problems, caesar shift
- Research: Vigenère cipher
- Bring your questions for Monday!



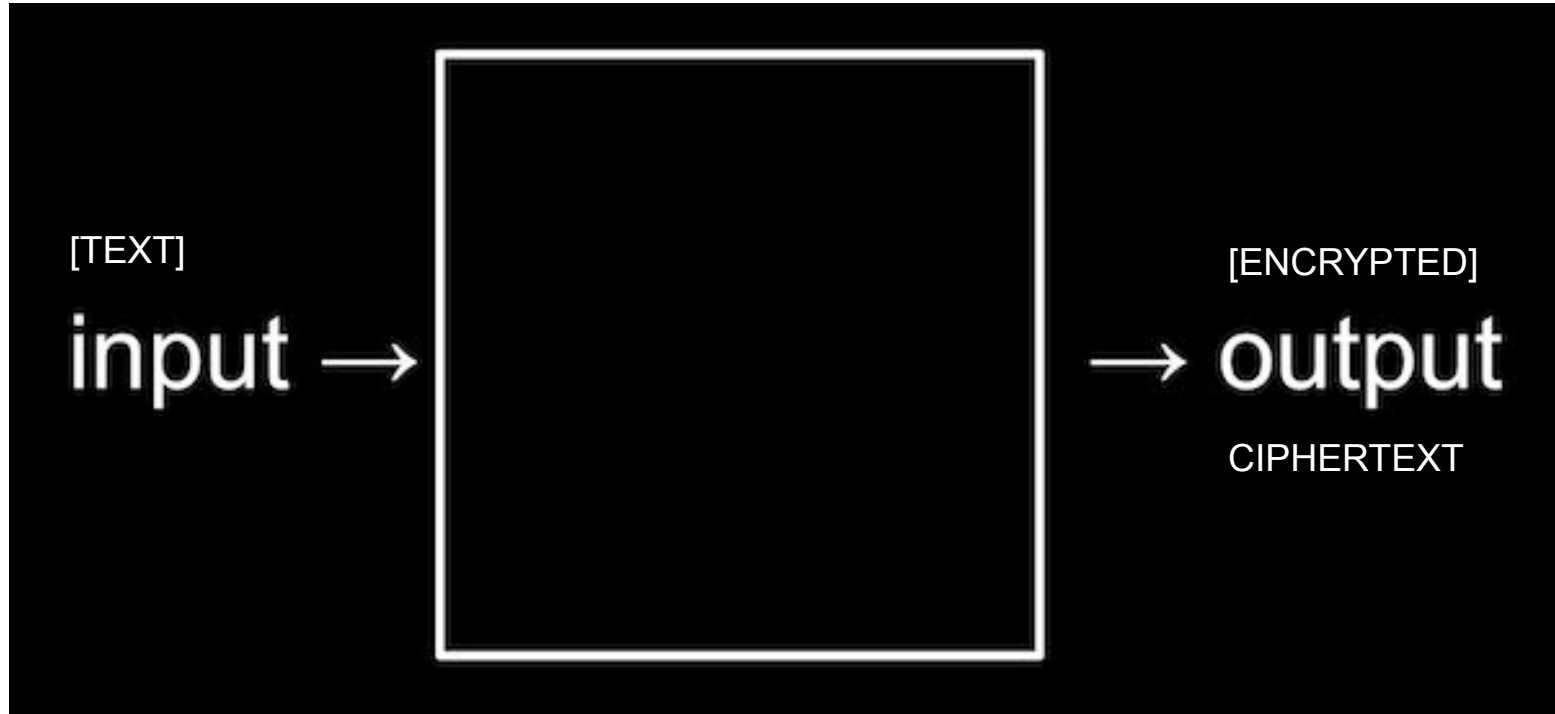
10.09 emoji



mail.at(0) mail.at(1) mail.at(2) mail.at(3) mail.at(4) mail.at(5)



cryptography = hidden writing (Greek)

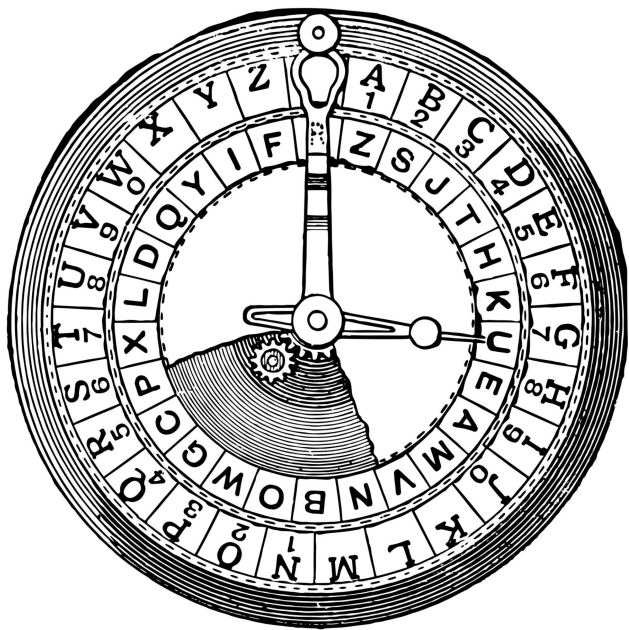


*From Harvard CS50*

# cipher, cypher

- Origins: meant "zero" or circle
- Then meant number, doing arithmetic
- Then meant **encoding** text, hidden

cipher, cypher





Caesar cipher

Caesar shift

Substitution  
cipher

military messages,  
1st cent. CE

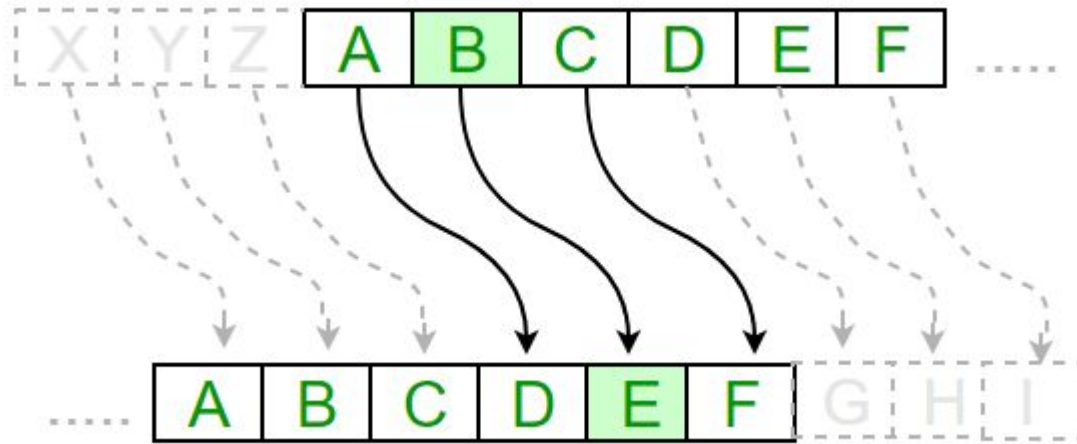




Original Message: "AT DAWN"


SHIFT, or KEY: 3

Encrypted Message: "DW GDZQ"





## Caesar shift in c++ :

- Tmp variables? Capital/lowercase?
-  shift as input, integer
- Decrypted ciphertext as output
- 2 vectors: 1 container for input text, 1 container for output text
- Nested loops: loop for the word, loop for the character; SHIFT EACH LETTER BY KEY, <INT>
- Alphabet table: ASCII!!

ENCRYPT

🔑 = 1

$$90 - 65 = 25$$

$$25 + \text{🔑} = 26$$

$$26 \% 26 = 0$$

$$0 + 65 = 65 = A$$

65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z
91	[

26 letters in  
alphabet

$$E_n(x) = (x + n) \mod 26.$$

DECRYPT

🔑 = 1

$$65 - 65 = 0$$

$$0 - \text{🔑} = -1$$

$$-1 + 26 = 25$$

$$25 \% 26 = 25$$

$$25 + 65 = 90 = Z$$

65	A
66	B
67	C
68	D
69	E
70	F
71	G
72	H
73	I
74	J
75	K
76	L
77	M
78	N
79	O
80	P
81	Q
82	R
83	S
84	T
85	U
86	V
87	W
88	X
89	Y
90	Z
91	[

26 letters in  
alphabet

**Alice and Bob**

Alice and Bob agree on a key  $K$

Secret key  $K$

**Alice**

Alice encrypts a message  $M$  using the key  $K$  and sends the encrypted message  $X$  to Bob.

Secret key  $K$

**Bob**

Bob receives the encrypted message  $X$  and decrypts  $X$  using the key  $K$  to obtain  $M$ .

$X$



OH *ALICE*... YOU'RE  
THE ONE FOR ME

BUT *BOB*... IN A  
*QUANTUM* WORLD  
HOW CAN WE BE SURE?

$\psi^+$  or  $\psi^-$ ?

THE APP WILL LET YOU SEND  
MESSAGES TO YOUR FRIEND  
ROBERT, OR MY BROTHER.

CAN THEY REPLY?

NO.

MY NEW SECURE TEXTING APP  
ONLY ALLOWS PEOPLE NAMED  
ALICE TO SEND MESSAGES  
TO PEOPLE NAMED BOB.

vectors vs. arrays?



```
vector<int> myNumbers (20) ;
```

```
int myNumbers [20] ;
```

## vectors vs. arrays

- **BIG DIFFERENCE:**  
**YOU CANNOT SIMPLY RESIZE ARRAYS!**
- This makes arrays faster, if you are *really* in need of speedy performance (large, large datasets)
- Arrays don't need a header `#include`
- For our purposes, simpler to use dynamic vectors

`myContainer.at(i)` vs. `myContainer[i]`

- `.at()` function checks the size of your container
- `[]` does not check the range!
- Both work for vectors
- Only `[]` works for built-in arrays ... BUT ...

## C-strings vs. strings (C++)

```
char myWord[6] = "hello";  
string myWord = "hello";
```

- C-string = older, from C, built-in
- Literally array of characters, with `'\0'` to **END** (size = +1)
- Different library of functions than C++ strings
- Easy to make mistakes with!

## Why learn arrays + C-strings?

- We still see them in code ("legacy")
- Good to understand the most basic data types (like ... binary) to know where our more advanced features come from!
- Vectors came from limitations with arrays, the standard template string from limitations with C-strings, etc.!