CS 103000
Prof. Madeline Blount

Week 7:
VECTORS (cont.)

Attendance:

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

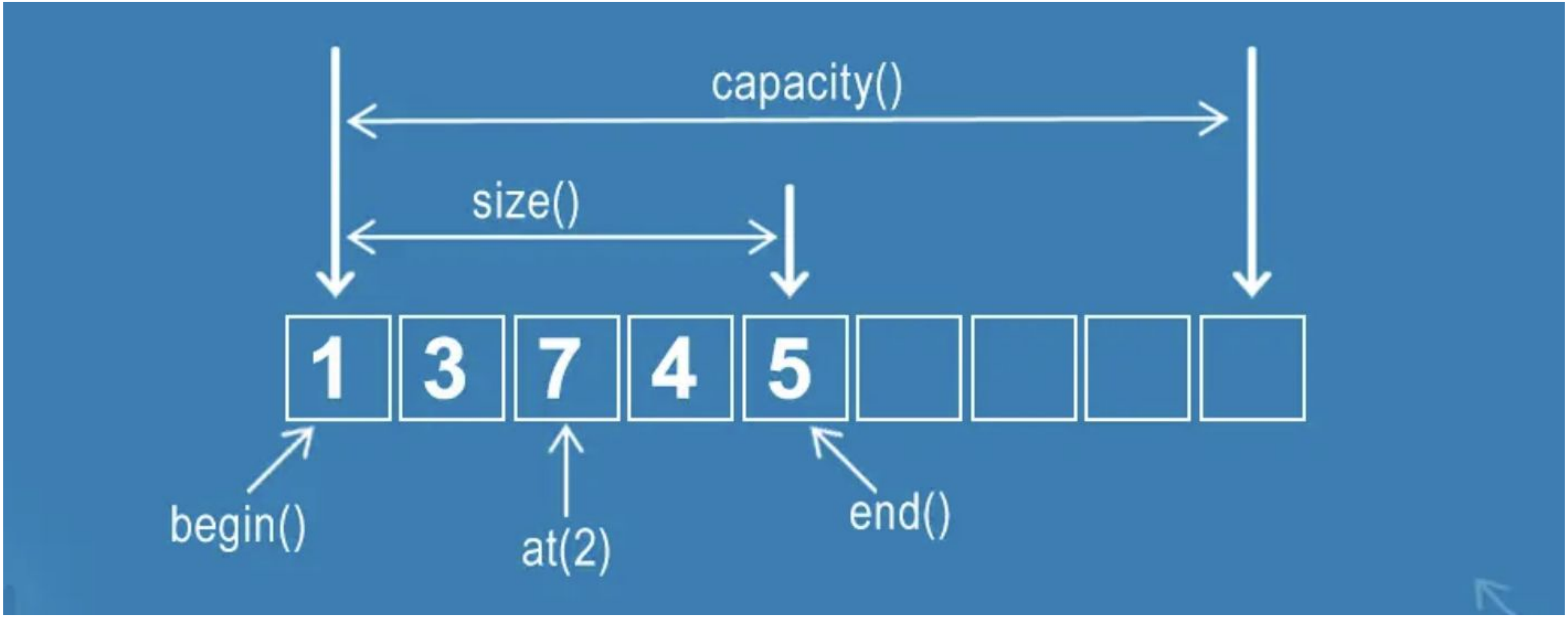
Dall-E 2: cats learning C++ in the forest on ‘90’s technology

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

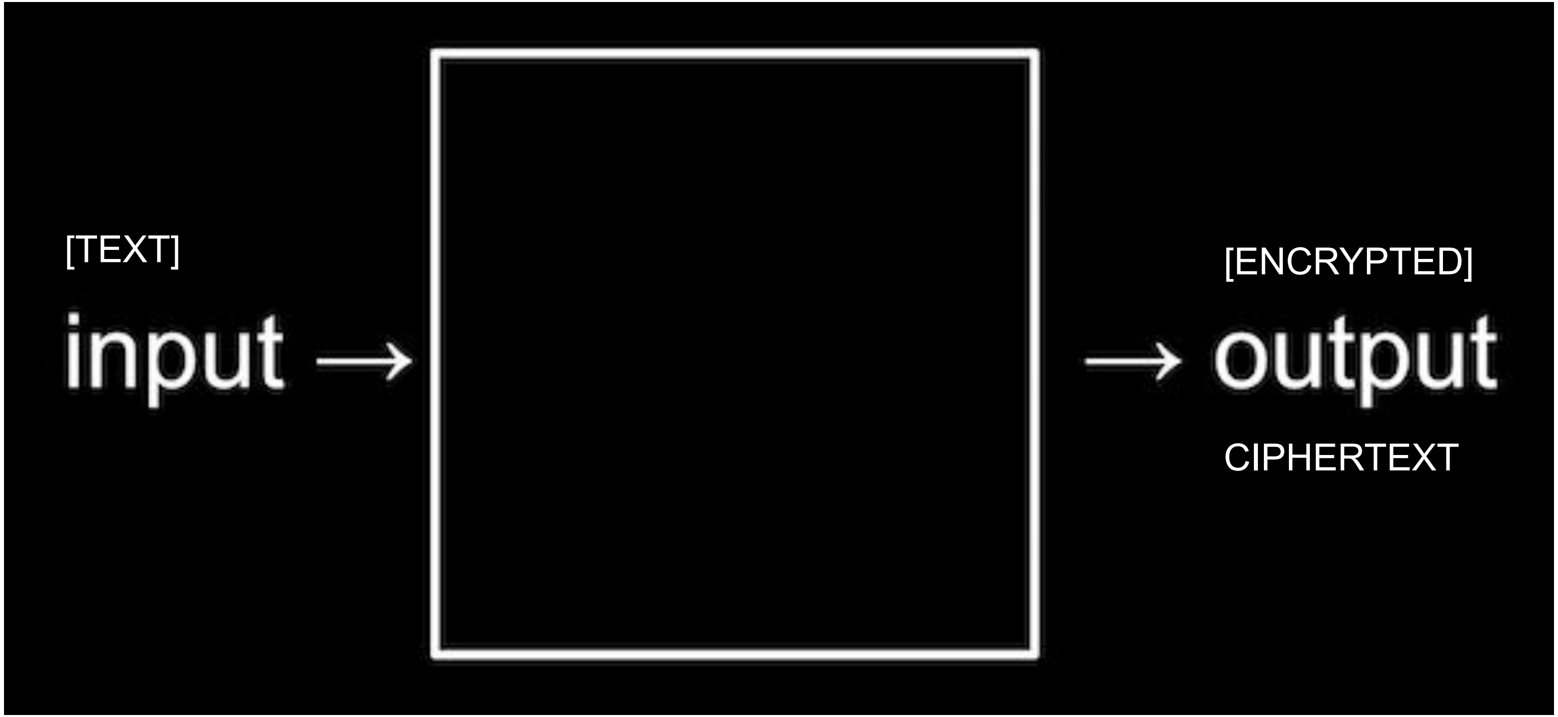
mail.at(0) mail.at(1) mail.at(2) mail.at(3) mail.at(4) mail.at(5)

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

vectors!



```
cryptography = hidden writing (Greek)
```

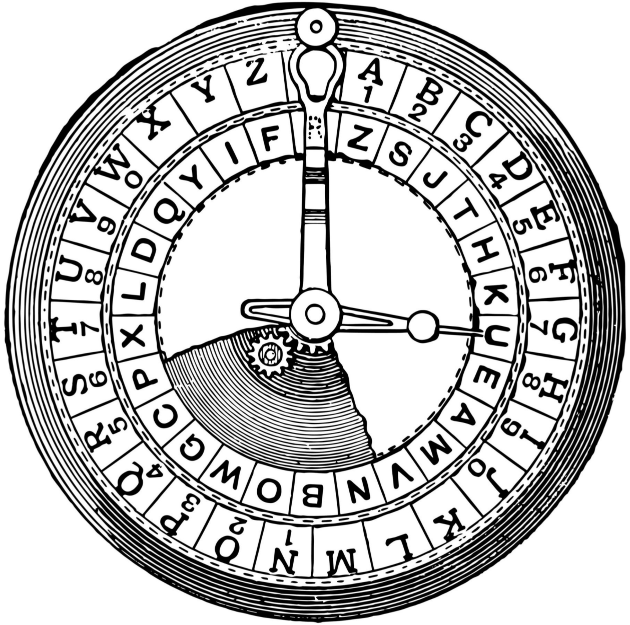


From Harvard CS50

cipher, cypher

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

cipher, cypher



Caesar cipher

Caesar shift

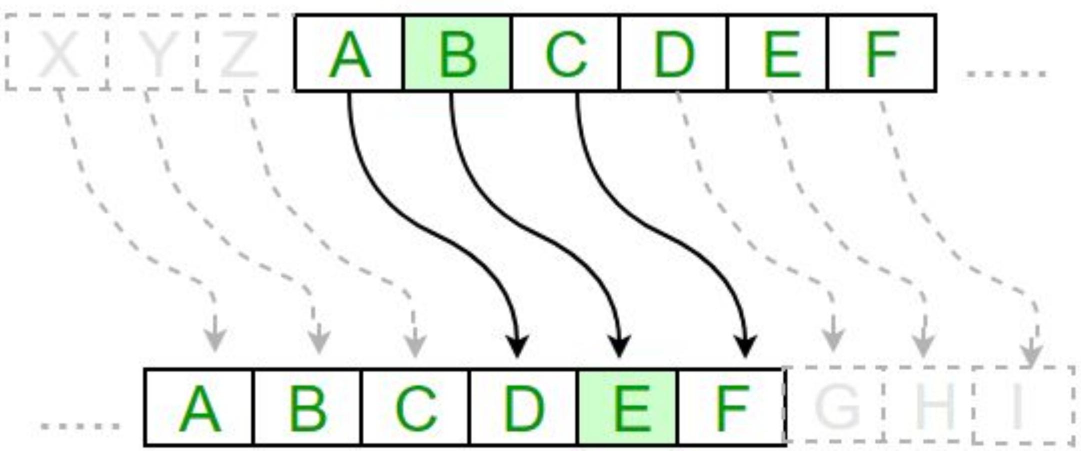
Substitution cipher

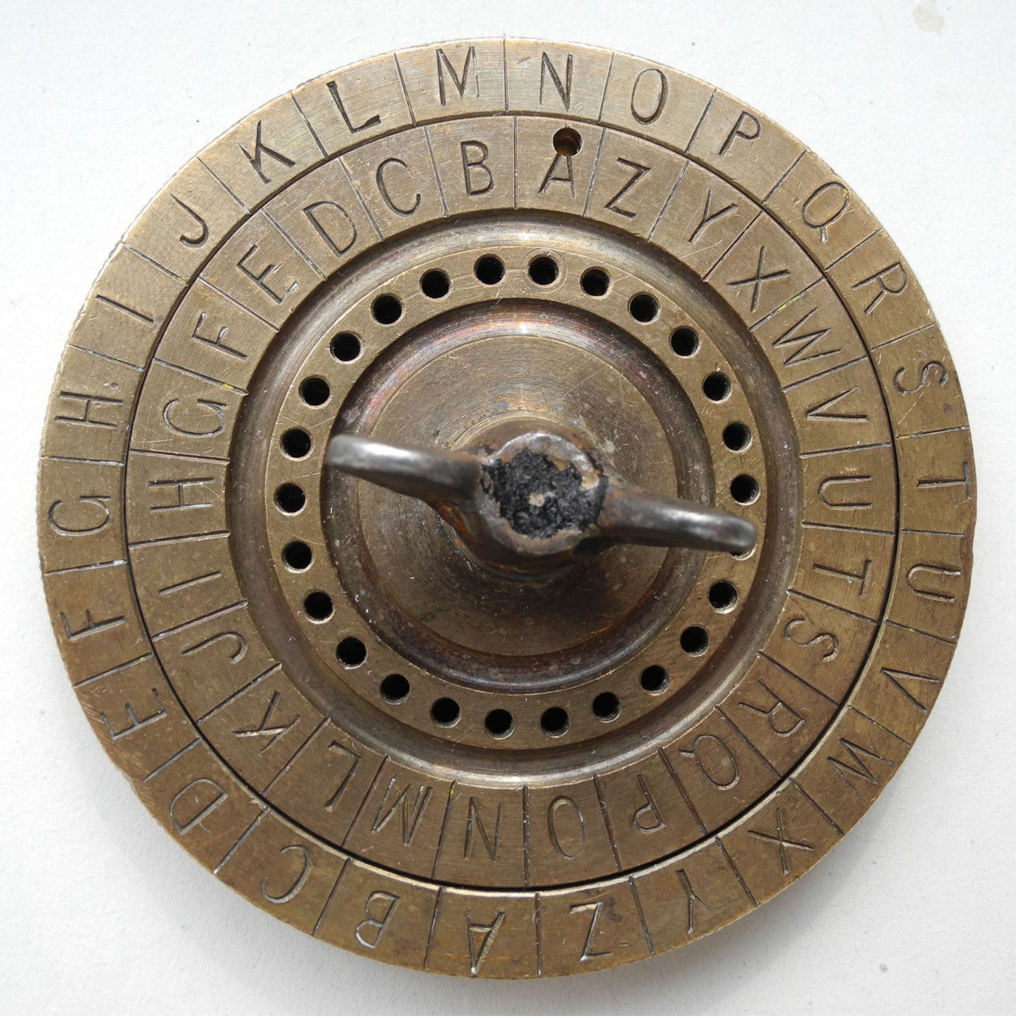
military messages, 1st cent. CE

Original Message: “ATTACK AT DAWN”

SHIFT, or KEY: 3

Encrypted Message: “DWWDFN DW GDZQ”




A B C D E F G H. I J K L M N O P Q R S T U V W X Y Z
A Z Y X W V U T S R Q P O N M L K J I H G F E D C B

Wikimedia

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

Caesar shift in c++ :

- String as input
- 🔑 shift as input
- Decrypted ciphertext as output

ENCRYPT

🔑 = 1

90 - 65 = 25

25 + 🔑 = 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 - 🔑 = -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

vectors vs. arrays?

Corporate needs you to find the differences
between this picture and this picture.
They're the same picture.

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)

- 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.!