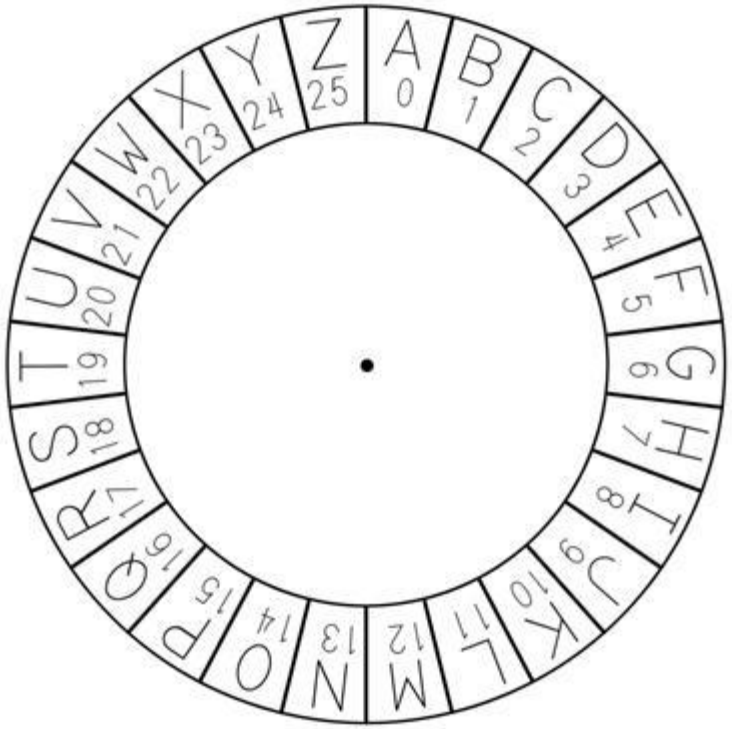
A Caesar cipher wheel

Your secret key is a number ƒ between 1 and 25.

To encrypt a letter, start at that letter and then count ƒ steps clockwise; where you land is your ciphertext.

Or:

To encrypt a letter, find its number, then add ƒ. If the result is greater than or equal to 26, subtract 26. (This is called computing the sum **mod 26**.) The ciphertext is the letter corresponding to the answer.



Example: If ƒ =7 then the letter M is encrypted as T because M=12 and 12+7 = 19 = T.

Example: if ƒ =17 then the letter M is encrypted as D because M=12 and 12+17=29 and 29-26=3 = D.

*In general: doing arithmetic mod 26 means : do the math, then: divide by 26 and just keep the remainder. The remainder of dividing 29 by 26 is 3. You can do more interesting things: 7 times 9 is 63, so mod 26 our answer is 11.

uOttawa

Binary numbers magic trick.

Choose your secret number between 0 and 127. Write “1” below each card that contains your number; write “0” below each card that doesn’t contain your number. Give that sequence of 0s and 1s to Eve/Yves, who will magically deduce your secret number.

Card 5			
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63

Card 4			
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63

Card 3			
8	9	10	11
12	13	14	15
24	25	26	27
28	29	30	31
40	41	42	43
44	45	46	47
56	57	58	59
60	61	62	63

Card 2			
4	5	6	7
12	13	14	15
20	21	22	23
28	29	30	31
36	37	38	39
44	45	46	47
52	53	54	55
60	61	62	63

Card 1			
2	3	6	7
10	11	14	15
18	19	22	23
26	27	30	31
34	35	38	39
42	43	46	47
50	51	54	55
58	59	62	63

Card 0			
1	3	5	7
9	11	13	15
17	19	21	23
25	27	29	31
33	35	37	39
41	43	45	47
49	51	53	55
57	59	61	63

1 if here; 0 if not	**1 if here; 0 if not**	**1 if here; 0 if not**	**1 if here; 0 if not**	**1 if here; 0 if not**	**1 if here; 0 if not**	**Ans**

uOttawa

UTF-8 encoding chart

(http://www.utf8-chartable.de/unicode-utf8-table.pl)

Char	Code	Char	Code	Char	Code	Char	Code
!	33	1	49	A	65	Q	81
"	34	2	50	B	66	R	82
#	35	3	51	C	67	S	83
$	36	4	52	D	68	T	84
%	37	5	53	E	69	U	85
&	38	6	54	F	70	V	86
'	39	7	55	G	71	W	87
(	40	8	56	H	72	X	88
)	41	9	57	I	73	Y	89
*	42	:	58	J	74	Z	90
+	43	;	59	K	75	[	91
,	44	<	60	L	76	\	92
-	45	=	61	M	77	]	93
.	46	>	62	N	78	^	94
/	47	?	63	O	79	_	95
0	48	@	64	P	80	`	96

UTF-8 actually encodes 2,164,864 different characters – covering many languages and alphabets, as well as fun symbols, such as:

γ	206 179	Ŧ	197 166	ɑ	198 163	Ꙗ	209 168
δ	206 180	ŧ	197 167	Ꝑ	198 164	ꙗ	209 169
ε	206 181	Ũ	197 168	ƥ	198 165	Ѫ	209 170

Binary numbers chart

1	=	1	= 1
10	=	2+0	= 2
11	=	2+1	= 3
100	=	4+0+0	= 4
101	=	4+0+1	= 5
110	=	4+2+0	= 6
111	=	4+2+1	= 7
1000	=	8+0+0+0	= 8
1001	=	8+0+0+1	= 9
1010	=	8+0+2+0	= 10
1011	=	8+0+2+1	= 11
1100	=	8+4+0+0	= 12
1101	=	8+4+0+1	= 13
1110	=	8+4+2+0	= 14
1111	=	8+4+2+1	= 15
10000	=	16+0+0+0+0	= 16
10001	=	16+0+0+0+1	= 17
10010	=	16+0+0+2+0	= 18
10011	=	16+0+0+2+1	= 19
10100	=	16+0+4+0+0	= 20
10101	=	16+0+4+0+1	= 21
10110	=	16+0+4+2+0	= 22
10111	=	16+0+4+2+1	= 23
11000	=	16+8+0+0+0	= 24
11001	=	16+8+0+0+1	= 25
11010	=	16+8+0+2+0	= 26
11011	=	16+8+0+2+1	= 27
11100	=	16+8+4+0+0	= 28
11101	=	16+8+4+0+1	= 29
11110	=	16+8+4+2+0	= 30
11111	=	16+8+4+2+1	= 31

uOttawa

ASCII character table

Decimal	Hexadecimal	Binary	Octal	Char
0	0	0	0	[NULL]
1	1	1	1	[START OF HEADING]
2	2	10	2	[START OF TEXT]
3	3	11	3	[END OF TEXT]
4	4	100	4	[END OF TRANSMISSION]
5	5	101	5	[ENQUIRY]
6	6	110	6	[ACKNOWLEDGE]
7	7	111	7	[BELL]
8	8	1000	10	[BACKSPACE]
9	9	1001	11	[HORIZONTAL TAB]
10	A	1010	12	[LINE FEED]
11	B	1011	13	[VERTICAL TAB]
12	C	1100	14	[FORM FEED]
13	D	1101	15	[CARRIAGE RETURN]
14	E	1110	16	[SHIFT OUT]
15	F	1111	17	[SHIFT IN]
16	10	10000	20	[DATA LINK ESCAPE]
17	11	10001	21	[DEVICE CONTROL 1]
18	12	10010	22	[DEVICE CONTROL 2]
19	13	10011	23	[DEVICE CONTROL 3]
20	14	10100	24	[DEVICE CONTROL 4]
21	15	10101	25	[NEGATIVE ACKNOWLEDGE]
22	16	10110	26	[SYNCHRONOUS IDLE]
23	17	10111	27	[ENG OF TRANS. BLOCK]
24	18	11000	30	[CANCEL]
25	19	11001	31	[END OF MEDIUM]
26	1A	11010	32	[SUBSTITUTE]
27	1B	11011	33	[ESCAPE]
28	1C	11100	34	[FILE SEPARATOR]
29	1D	11101	35	[GROUP SEPARATOR]
30	1E	11110	36	[RECORD SEPARATOR]
31	1F	11111	37	[UNIT SEPARATOR]
32	20	100000	40	[SPACE]
33	21	100001	41	!
34	22	100010	42	"
35	23	100011	43	#
36	24	100100	44	$
37	25	100101	45	%
38	26	100110	46	&
39	27	100111	47	'
40	28	101000	50	(
41	29	101001	51	)
42	2A	101010	52	*
43	2B	101011	53	+
44	2C	101100	54	,
45	2D	101101	55	-
46	2E	101110	56	.
47	2F	101111	57	/

Decimal	Hexadecimal	Binary	Octal	Char
48	30	110000	60	0
49	31	110001	61	1
50	32	110010	62	2
51	33	110011	63	3
52	34	110100	64	4
53	35	110101	65	5
54	36	110110	66	6
55	37	110111	67	7
56	38	111000	70	8
57	39	111001	71	9
58	3A	111010	72	:
59	3B	111011	73	;
60	3C	111100	74	<
61	3D	111101	75	=
62	3E	111110	76	>
63	3F	111111	77	?
64	40	1000000	100	@
65	41	1000001	101	A
66	42	1000010	102	B
67	43	1000011	103	C
68	44	1000100	104	D
69	45	1000101	105	E
70	46	1000110	106	F
71	47	1000111	107	G
72	48	1001000	110	H
73	49	1001001	111	I
74	4A	1001010	112	J
75	4B	1001011	113	K
76	4C	1001100	114	L
77	4D	1001101	115	M
78	4E	1001110	116	N
79	4F	1001111	117	O
80	50	1010000	120	P
81	51	1010001	121	Q
82	52	1010010	122	R
83	53	1010011	123	S
84	54	1010100	124	T
85	55	1010101	125	U
86	56	1010110	126	V
87	57	1010111	127	W
88	58	1011000	130	X
89	59	1011001	131	Y
90	5A	1011010	132	Z
91	5B	1011011	133	[
92	5C	1011100	134	\
93	5D	1011101	135	]
94	5E	1011110	136	^
95	5F	1011111	137	_

Decimal	Hexadecimal	Binary	Octal	Char
96	60	1100000	140	`
97	61	1100001	141	a
98	62	1100010	142	b
99	63	1100011	143	c
100	64	1100100	144	d
101	65	1100101	145	e
102	66	1100110	146	f
103	67	1100111	147	g
104	68	1101000	150	h
105	69	1101001	151	i
106	6A	1101010	152	j
107	6B	1101011	153	k
108	6C	1101100	154	l
109	6D	1101101	155	m
110	6E	1101110	156	n
111	6F	1101111	157	o
112	70	1110000	160	p
113	71	1110001	161	q
114	72	1110010	162	r
115	73	1110011	163	s
116	74	1110100	164	t
117	75	1110101	165	u
118	76	1110110	166	v
119	77	1110111	167	w
120	78	1111000	170	x
121	79	1111001	171	y
122	7A	1111010	172	z
123	7B	1111011	173	{
124	7C	1111100	174	|
125	7D	1111101	175	}
126	7E	1111110	176	~
127	7F	1111111	177	[DEL]

The language of computers is **binary**.

uOttawa

Devising a method for secret key exchange in a completely insecure environment

Goal:

Devise a means for Alice to send a **secret key** to Bob **even though** Eve can intercept (but not break) **everything** that Alice gives to Bob.

Items: (use some, or all, of them)

Tool box (empty)	Plank
Rope	Light switch
Lightbulb	Refrigerator
Two locks	Elephant
Keys for the locks	Feather

Three teams can present their solutions after lunch.

Calculating powers of 2 mod 11

Fill in the table below.

Multiply the number in each line by 2, then divide by 11 and record the remainder on the next line.

Continue until you get a cycle.

n	**2^n**	**$[2^n]_{11}$ = remainder when you divide 2^n by 11**
0	1	1
1	2	2
2	4	4
3	8	8
4	16	5
5	32	
6	64	
7		
8		
9		
10		
11		
12		

uOttawa

Diffie-Hellman key exchange algorithm

Alice	**Bob**
Choose a prime number p and a base x.	
Choose a secret number a between 2 and p-1. *Tip: make sure that $x^a > p$.*	Choose a secret number b. *Tip: make sure that $x^b > p$.*
Calculate x to the power a mod p: $[x^a]_p$. Call the answer A. *Tip: use an online modular exponentiation calculator.*	Calculate x to the power b mod p: $[x^b]_p$. Call the answer B.
Alice gives A to Bob. Bob gives B to Alice.	
Calculate B to the power a mod p : $[B^a]_p$. Call it K.	Calculate A to the power b mod p : $[A^b]_p$. Call it K.
The reveal: compare your answers. The number K is your shared secret key.	

Example:

- Alice and Bob choose p=541 and x=2.
- Alice picks a=100, and uses a calculator to find that $A=[2^{100}]_{541}=2^{100} \pmod{541} = 34$.
- Separately and secretly, Bob picks his secret number b and finds B = 74.
- Then Alice takes Bob's answer, 74, to her secret power a=100, mod 541, and gets the answer K=118.
- Meanwhile, Bob's secret number was b=100; he takes Alice's public 34 to his secret power 200 (mod 541), and amazingly also gets K=118.

The secret is: $(2^{100})^{200} = 2^{20000} = (2^{200})^{100}$ so they have the same remainder when you divide by p=541.

Some useful links

- Lists of large prime numbers: BigPrimes.net http://www.bigprimes.net/archive/prime/
- A large codes and cryptography website dcode.fr which has:
  - An ASCII converter (between characters and numbers written in decimal or binary) http://www.dcode.fr/ascii-code
  - A modular exponentiation calculator http://www.dcode.fr/modular-exponentiation-calculus
- Another big cryptography resource is Bill's ASecuritySite.com, which has:
  - A random prime generator, to create primes of any length https://asecuritysite.com/encryption/random3
  - A Diffie-Hellman calculator, that lets you play both Alice and Bob https://asecuritysite.com/encryption/diffie
- Another modular exponentiation calculator : http://comnuan.com/cmnn02/cmnn02008/
- Activities for classrooms with cryptography http://crypto.interactive-maths.com/downloadable-resources.html
- Some great authors: Tattersall (An introduction to number theory in nine chapters) and Simon Singh (The Code Book).
- My website, with a copy of the presentation and handouts: http://mysite.science.uottawa.ca/mnevins/talks.html#pub

Monica Nevins

mnevins@uottawa.ca