



comp1511 week 7

starting 5 minutes past the hour

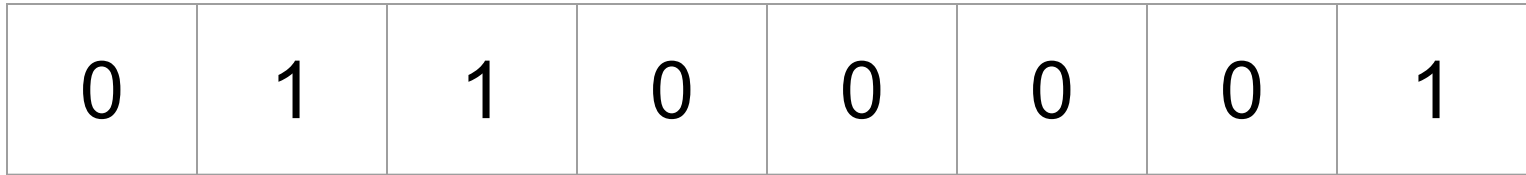
notices

- assignment 1 is done!
 - congratulations :D

today

- characters and strings
 - `getchar()`, `putchar()`
 - `fgets()`
- command line arguments
- testing
- bonus crypto stuff if you're interested

what is a char?



|————— 1 byte —————|

ascii table

0	<NUL>	32	<SPC>	64	@	96	`	128	À	160	†	192	ˆ	224	+
1	<SOH>	33	!	65	A	97	a	129	Á	161	°	193	ı	225	·
2	<STX>	34	"	66	B	98	b	130	Â	162	¢	194	ı	226	,
3	<ETX>	35	#	67	C	99	c	131	Ã	163	£	195	√	227	„
4	<EOT>	36	\$	68	D	100	d	132	Ä	164	§	196	f	228	%
5	<ENQ>	37	%	69	E	101	e	133	Å	165	•	197	≈	229	Â
6	<ACK>	38	&	70	F	102	f	134	Ö	166	¶	198	Δ	230	Ê
7	<BEL>	39	'	71	G	103	g	135	á	167	Ø	199	«	231	Á
8	<BS>	40	(72	H	104	h	136	â	168	®	200	»	232	Ê
9	<TAB>	41)	73	I	105	i	137	ã	169	©	201	...	233	Ë
10	<LF>	42	*	74	J	106	j	138	ä	170	™	202		234	İ
11	<VT>	43	+	75	K	107	k	139	å	171	·	203	À	235	Í
12	<FF>	44	,	76	L	108	l	140	â	172	ˆ	204	Ã	236	Î
13	<CR>	45	-	77	M	109	m	141	ç	173	≠	205	Ö	237	İ
14	<SO>	46	.	78	N	110	n	142	é	174	Æ	206	œ	238	Ó
15	<SI>	47	/	79	O	111	o	143	è	175	ø	207	œ	239	Ô
16	<DL>	48	0	80	P	112	p	144	ê	176	∞	208	-	240	■
17	<DC1>	49	1	81	Q	113	q	145	ë	177	±	209	-	241	Ò
18	<DC2>	50	2	82	R	114	r	146	í	178	≤	210	"	242	Ó
19	<DC3>	51	3	83	S	115	s	147	ì	179	≥	211	"	243	Ô
20	<DC4>	52	4	84	T	116	t	148	ï	180	¥	212	'	244	Ù
21	<NAK>	53	5	85	U	117	u	149	İ	181	µ	213	'	245	ı
22	<SYN>	54	6	86	V	118	v	150	ñ	182	ø	214	÷	246	ˆ
23	<ETB>	55	7	87	W	119	w	151	ó	183	Σ	215	◊	247	˜
24	<CAN>	56	8	88	X	120	x	152	ò	184	Π	216	ÿ	248	-
25		57	9	89	Y	121	y	153	ó	185	n	217	Ÿ	249	˘
26	<SUB>	58	:	90	Z	122	z	154	ø	186	ƒ	218	/	250	·
27	<ESC>	59	;	91	[123	{	155	ø	187	ª	219	€	251	°
28	<FS>	60	<	92	\	124		156	ú	188	º	220	<	252	ˆ
29	<GS>	61	=	93]	125	}	157	û	189	Ω	221	>	253	˜
30	<RS>	62	>	94	^	126	~	158	ü	190	æ	222	fi	254	˘
31	<US>	63	?	95	_	127		159	ü	191	ø	223	fi	255	˙

strings

'w'	'o'	'w'	'!'	'\n'	'\0'	'%'	'&'
-----	-----	-----	-----	------	------	-----	-----



NULL
terminator

some special functions

this	is very similar to
<pre>int ch; ch = getchar();</pre>	<pre>int ch; scanf("%c", &ch);</pre>
<pre>char string[100]; fgets(string, 100, stdin);</pre>	<pre>char string[100]; scanf("%s", string);</pre>

7. Write a program `sum_digits.c` which reads characters from its input. When the end of input is reached it should print a count of the number of digits in its input and their sum.

The only functions you can use are `getchar()` and `printf()`.

For example:

```
$ ./sum_digits
1 2 3 o'clock
4 o'clock rock
Ctrl-D
Input contained 4 digits which summed to 10
$ ./sum_digits
12 twelve 24 twenty four
thirty six 36
Ctrl-D
Input contained 6 digits which summed to 18
```


fgets

Description

The C library function **char *fgets(char *str, int n, FILE *stream)** reads a line from the specified stream and stores it into the string pointed to by **str**. It stops when either **(n-1)** characters are read, the newline character is read, or the end-of-file is reached, whichever comes first.

Declaration

Following is the declaration for fgets() function.

```
char *fgets(char *str, int n, FILE *stream)
```

Parameters

- **str** – This is the pointer to an array of chars where the string read is stored.
- **n** – This is the maximum number of characters to be read (including the final null-character). Usually, the length of the array passed as str is used.
- **stream** – This is the pointer to a FILE object that identifies the stream where characters are read from.

Return Value

On success, the function returns the same str parameter. If the End-of-File is encountered and no characters have been read, the contents of str remain unchanged and a null pointer is returned.

If an error occurs, a null pointer is returned.

https://www.tutorialspoint.com/c_standard_library/c_function_fgets.htm

what is stored in argc and argv?

`./add 10 20 30`

testing

why is it important?

what should you test for?

week 7 survey

<https://forms.gle/8S4viHumtZNWb8xF6>

How confident do you feel with the content in the course so far?

- ☐ Pretty good :D
- ☐ Good - mostly on top of things :)
- ☐ Not bad - a bit behind
- ☐ Pretty lost
- ☐ Other: _____

What do you find most helpful in the tutorials? *

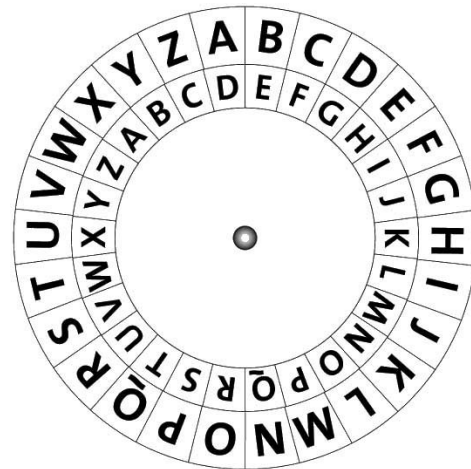
- ☐ Kahoots
- ☐ Lecture Revision
- ☐ Interactive (code/draw/discuss in small groups)
- ☐ Code Examples
- ☐ Deep diving into lecture content
- ☐ Other: _____

Anything else you want to say?

Your answer _____

intro to crypto (bonus slides)

caesar cipher



cyclically shift each letter k places forward

$k = 3$

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

For $k = 3$, the plaintext **HELLO** is encrypted as **KHOOR**

simple substitution cipher

permute the alphabet for a key, then map letters to encrypt.

mapped alphabet to a scrambled version

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

The plaintext **HELLO** is encrypted as **XUOOB**

number of keys

$$|K| = 26! \approx 4 \times 10^{26}$$

that's a big number!!!

decryption - the magic of frequency

