CS 31 Discussion

ABDULLAH-AL-ZUBAER IMRAN

WEEK 5: ARRAYS

Recap

- Strings
  - Characters and Strings

- Functions
  - Built-in vs user-defined
  - Function declaration, definition, and call
  - Parameter passing: Pass by value
    Pass by reference

Good Coding Practices

- Have descriptive variable names
- Use indentation to clarify meaning
- Have short, descriptive comments
- Incremental development

Incremental development tips

- Write small blocks of code at a time and test them with multiple types of input

- If you have multiple errors, test them sequentially and recompile as you fix mistakes. Sometimes fixing one mistake will also fix later ones.

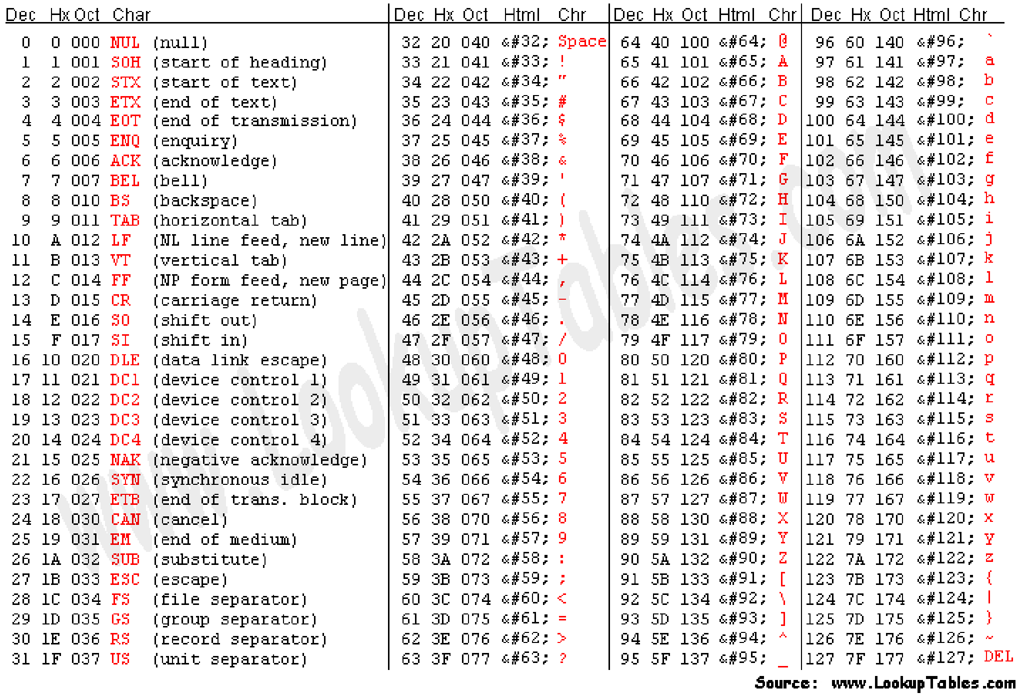
- Save often!

Convert letter to integer

Define a function

```
int convert(char number)
{
        switch (number)
        {
                case '0':
                   return 0;
                case '1':
                   return 1;
                case '2':
                   return 2;
                case '3':
                   return 3;
                case '4':
                   return 4;
                case '5':
                   return 5;
                case '6':
                   return 6;
                case '7':
                   return 7;
                case '8':
                   return 8;
                case '9':
                   return 9;
        }
}
```

Characters and Integers



Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	&#32;	Space	64	40	100	&#64;	@	96	60	140	&#96;	`
1	1	001	SOH (start of heading)	33	21	041	&#33;	!	65	41	101	&#65;	A	97	61	141	&#97;	a
2	2	002	STX (start of text)	34	22	042	&#34;	"	66	42	102	&#66;	B	98	62	142	&#98;	b
3	3	003	ETX (end of text)	35	23	043	&#35;	#	67	43	103	&#67;	C	99	63	143	&#99;	c
4	4	004	EOT (end of transmission)	36	24	044	&#36;	$	68	44	104	&#68;	D	100	64	144	&#100;	d
5	5	005	ENQ (enquiry)	37	25	045	&#37;	%	69	45	105	&#69;	E	101	65	145	&#101;	e
6	6	006	ACK (acknowledge)	38	26	046	&#38;	&	70	46	106	&#70;	F	102	66	146	&#102;	f
7	7	007	BEL (bell)	39	27	047	&#39;	'	71	47	107	&#71;	G	103	67	147	&#103;	g
8	8	010	BS (backspace)	40	28	050	&#40;	(	72	48	110	&#72;	H	104	68	150	&#104;	h
9	9	011	TAB (horizontal tab)	41	29	051	&#41;	)	73	49	111	&#73;	I	105	69	151	&#105;	i
10	A	012	LF (NL line feed, new line)	42	2A	052	&#42;	*	74	4A	112	&#74;	J	106	6A	152	&#106;	j
11	B	013	VT (vertical tab)	43	2B	053	&#43;	+	75	4B	113	&#75;	K	107	6B	153	&#107;	k
12	C	014	FF (NP form feed, new page)	44	2C	054	&#44;	,	76	4C	114	&#76;	L	108	6C	154	&#108;	l
13	D	015	CR (carriage return)	45	2D	055	&#45;	-	77	4D	115	&#77;	M	109	6D	155	&#109;	m
14	E	016	SO (shift out)	46	2E	056	&#46;	.	78	4E	116	&#78;	N	110	6E	156	&#110;	n
15	F	017	SI (shift in)	47	2F	057	&#47;	/	79	4F	117	&#79;	O	111	6F	157	&#111;	o
16	10	020	DLE (data link escape)	48	30	060	&#48;	0	80	50	120	&#80;	P	112	70	160	&#112;	p
17	11	021	DC1 (device control 1)	49	31	061	&#49;	1	81	51	121	&#81;	Q	113	71	161	&#113;	q
18	12	022	DC2 (device control 2)	50	32	062	&#50;	2	82	52	122	&#82;	R	114	72	162	&#114;	r
19	13	023	DC3 (device control 3)	51	33	063	&#51;	3	83	53	123	&#83;	S	115	73	163	&#115;	s
20	14	024	DC4 (device control 4)	52	34	064	&#52;	4	84	54	124	&#84;	T	116	74	164	&#116;	t
21	15	025	NAK (negative acknowledge)	53	35	065	&#53;	5	85	55	125	&#85;	U	117	75	165	&#117;	u
22	16	026	SYN (synchronous idle)	54	36	066	&#54;	6	86	56	126	&#86;	V	118	76	166	&#118;	v
23	17	027	ETB (end of trans. block)	55	37	067	&#55;	7	87	57	127	&#87;	W	119	77	167	&#119;	w
24	18	030	CAN (cancel)	56	38	070	&#56;	8	88	58	130	&#88;	X	120	78	170	&#120;	x
25	19	031	EM (end of medium)	57	39	071	&#57;	9	89	59	131	&#89;	Y	121	79	171	&#121;	y
26	1A	032	SUB (substitute)	58	3A	072	&#58;	:	90	5A	132	&#90;	Z	122	7A	172	&#122;	z
27	1B	033	ESC (escape)	59	3B	073	&#59;	;	91	5B	133	&#91;	[	123	7B	173	&#123;	{
28	1C	034	FS (file separator)	60	3C	074	&#60;	<	92	5C	134	&#92;	\	124	7C	174	&#124;	|
29	1D	035	GS (group separator)	61	3D	075	&#61;	=	93	5D	135	&#93;	]	125	7D	175	&#125;	}
30	1E	036	RS (record separator)	62	3E	076	&#62;	>	94	5E	136	&#94;	^	126	7E	176	&#126;	~
31	1F	037	US (unit separator)	63	3F	077	&#63;	?	95	5F	137	&#95;	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

ASCII code

char	int
‘0’	48
‘1’	49
‘2’	50
‘3’	51
‘4’	52
‘5’	53
‘6’	54
‘7’	55
‘8’	56
‘9’	57

‘0’ is not mapped to 0! However, the integer code for chars ‘0’ through ‘9’ are contiguous.

Characters and Integers

char	int
‘0’	48
‘1’	49
‘2’	50
‘3’	51
‘4’	52
‘5’	53
‘6’	54
‘7’	55
‘8’	56
‘9’	57

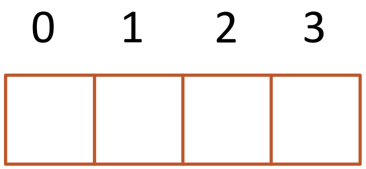
```
char ch = '0';
ch++; // ch is '1'
int a = ch - '0'; // a is 1
ch += 7; // ch is '8'
a = ch - '0'; // a is 8
```

Array

Declare an array

- <type> <name>[size]

```
int a[4];
```



a[i] is an i-th variable in the array a.

size should must be a positive integer constant.

```
int a[4];
```

```
const int N = 10;
int a[N];
```

You can treat each element of the array as a variable.

```
x[3] = 5;
x[1]++;
cout << x[i] << endl;
```

Initialization of an Array

```
int a[5] = {1, 2, 3, 5, 7};
int a[] = {1, 2, 3, 5, 7};
```

You cannot set the size to what’s less than the number of elements in the initialization statement.

However, it is okay to set the size to what’s more than the number of elements in the initialization statement.

```
int a[3] = {1, 2, 3, 5, 7}; // (right/wrong?)
int a[10] = {1, 2, 3, 5, 7}; // (right/wrong?)
```

Common mistakes

```
int a[10];
for (int i = 0; i < a.size(); i++) {
 ...
}
```

No `size()` function is defined for arrays.

```
const int SIZE = 10;
int a[SIZE];
for (int i = 0; i < SIZE; i++) {
 ...
}
```

Common mistakes

Out-of-range access not allowed!

```
int a[10];
a[15] = 5; // error
a[-10] = 4; // error
a[9] = 10; // okay
```

Arrays in a Function

```
void fibo10(int fib[]);
```

Note that the size of fib is not specified, you can explicitly pass the size in the function.

```
void fibo10(int fib[], int n);
```

Project3: Reminder

- Code:
  - Identifiers: meaningful variable and function names
  - Comments: meaningful comments

- Report
  - Pseudocode: Complete design of your program
    Outline the purpose of each line
  - Test cases: example cases with reasons

Thanks!

Questions?

Some of the materials presented have been taken from other TA discussions