

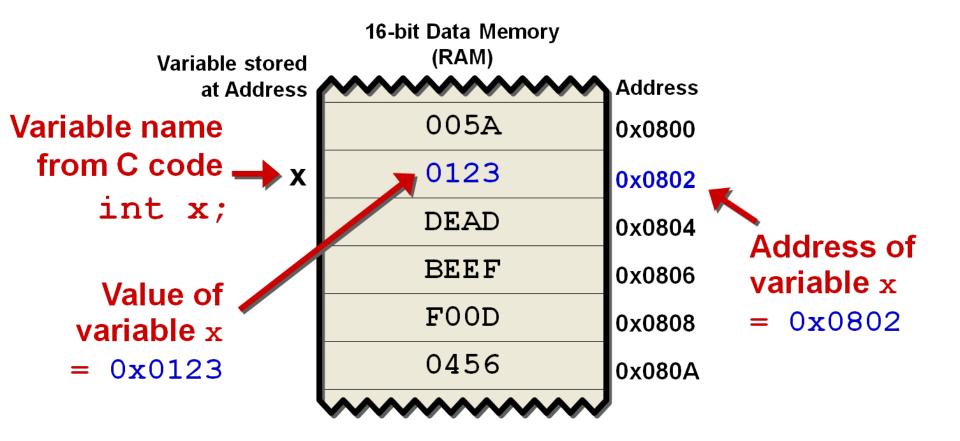
Introduction to C EECS 2031

Song Wang

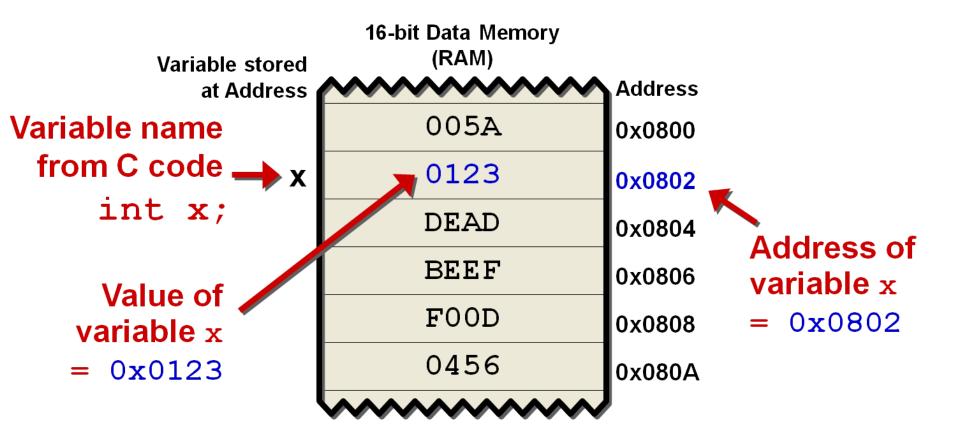
wangsong@eecs.yorku.ca eecs.yorku.ca/~wangsong/

Acknowledgement

- Some of the covered materials are based on previous EECS2031 offerings:
 - Uyen Trang (UT) Nguyen, Pooja Vashisth, Hui Wang, Manos Papagelis







printf(a): values of variables
scanf(&a): addresses of variables



#include <stdio.h>

```
int main(){
    float a, b;
    printf("Enter two float separated by <><>:");
    scanf("%f<><>%f", &a, &b);
    printf("these are %f <><>%f \n", a, b);
}
```

indigo 320 % ./read Enter two float separated by <><>:0.9<><>0.2 these are 0.900000 <><>0.200000 indigo 321 % ./read Enter two float separated by <><>:0.9<><>2 these are 0.900000 <><>2.000000



If we use "a" and "b" as the input for scanf():scanf("%f<><>%f", a, b);

#include <stdio.h>

```
int main() {
    float a, b;
    printf("Enter two float separated by <><>:");
    scanf("%f<><>%f", a, b);
    printf("these are %f <><>%f \n", a, b);
```

indigo 324 % gcc read2.c -o read2 indigo 325 % ./read2 Enter two float separated by <><>:0.9<><>0.2 Segmentation fault (core dumped)



If we use "&a" and "&b" as the input for
printf():printf("these are %f<><>%f", &a, &b);

#include <stdio.h>

```
int main() {
    float a, b;
    printf("Enter two float separated by <><>:");
    scanf("%f<><>%f", &a, &b);
    printf("these are %f <><>%f \n", &a, &b);
```

indigo 335 % gcc read3.c -o read3
indigo 336 % ./read3
Enter two float separated by <><>:0.9<><>0.2
these are 0.000000 <><>0.000000



- int getchar(void)
 - To read one character at a time from the standard input
 - Returns the next input char each time it is called;
 - Returns EOFwhen it encounters end of file.oend of file; Using < : end of input file
 - keyboard: Ctrl-D (Unix) or Ctrl-Z (Windows). "Keyboard is a file"
- EOF: an **int** constant defined in **<stdio.h**>, value is -1.

- •int putchar(int c)
 - Puts the character c on the standard output
 - Returns the character written (usually ignored);
 - Like printf("%c", c);



countChar.c

```
#include <stdio.h> // defines EOF
main() {
 int c;
 int count = 0;
 c = getchar();
 while(c != EOF) /* no end of file*/
   count++; //include spaces and '\n'
   c = getchar(); /* read next */
 printf("# of chars: %d\n",count);
```



countChar.c

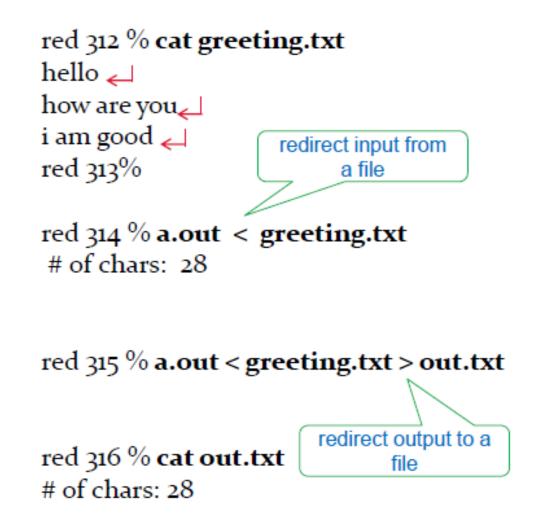
```
#include <stdio.h> // defines EOF
main() {
 int c;
 int count = 0;
 c = getchar();
 while(c != EOF) /* no end of file*/
   count++; //include spaces and '\n'
   c = getchar(); /* read next */
 printf("# of chars: %d\n",count);
```

red 309 % **a.out** hellc← how are vc← iam goor← Ctrl + D (end of the input)

of chars: 28



• Redirected from file





Statements

- Expression statement
 - y = i+1; i++; x = 4;
- Function call statement
 - printf("the result is %d");
- Control flow statement
 - if else, for(), while(), do while, case switch



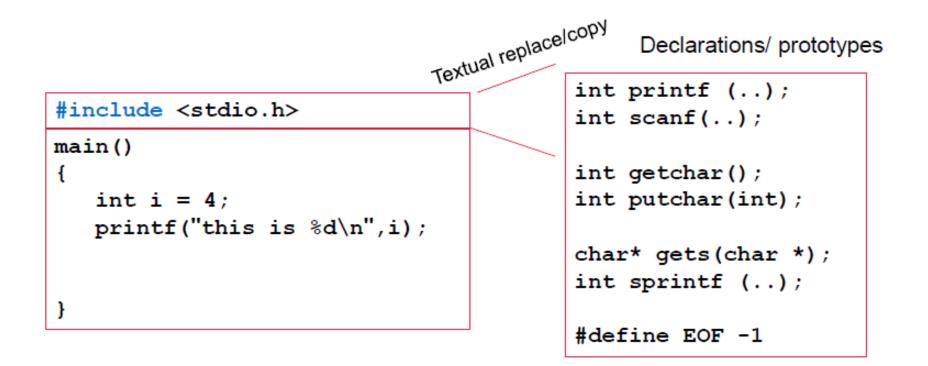
Expression

- Formed by combining operands(variable, constants and function calls) using operators (+ -* % > < == !=)
- Has return values
 - x+1
 - i < 20 false: 0 true: 1</p>
 - sum (i+j)
 - x = 5 = is an operator in C (and Java)! Return value 5
 - x = k + sum(i,j) printf("%d", x=5);



printf("%d", i<20);

Preprocessing: # include, #define

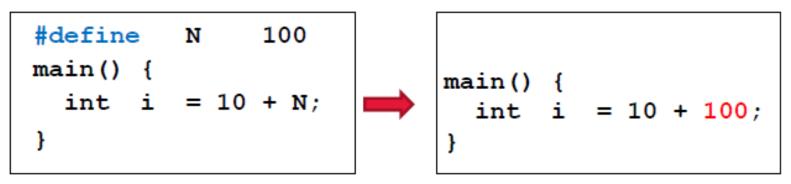




#define directive

• Syntax #define name value No type;

- Name called symbolic **constant**, conventionally written in upper case
- Value can be any sequence of characters



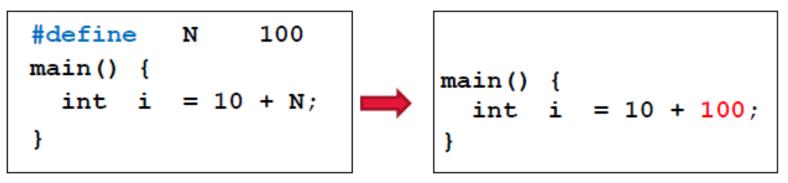
• Use as constant N = x + 2; X



#define directive

• Syntax #define name value No type;

- Name called symbolic **constant**, conventionally written in upper case
- Value can be any sequence of characters



• Use as constant N = x + 2; X
#define LENGTH 10
#define WIDTH 5
#define NEWLINE '\n'



C-Types, Operators, Expressions

- [Primitive/scalar] Types and sizes
 - Primitive Types
 - Constant values (literals)
- [Structured/aggregated] Array and "strings"
- Expressions
 - Basic operators
 - Type promotion and conversion
 - Other operators
 - Precedence of operators



C Primitive Types

- Variables and values have types
- There are two basic types in ANSI-C: integer, and floating point

Integer type

- char -character, 1 byte (8 bits)
- **short [int]** -short integer, usually2 bytes (16 bits)
- int -integer, usually2 or 4 bytes (16 or 32 bits)
- long [int] -long integer, usually4 or 8 bytes (32 or 64 bits)

• Floating point

- float
- double
- long double

-single-precision, usually4 bytes (32 bits)

- -double-precision, usually8 bytes (64 bits)
- -extended-precision



C Primitive Types and Sizes

• Variables and values have types

C Basis Data Types		32-bit CPU	64-bit CPU						
	size (bytes)	Range	size (bytes)	Range					
char	1	-128 to 128	1	-128 to 128					
short	2	-32,768 to 32,767	2	-32,768 to 32,767					
int	4	-2,147,483,648 to 2,147,483.647	4	-2,147,483,648 to 2,147,483.647					
long	4	-2,147,483,648 to 2.147,48.647	4	-2,147,483,648 to 2,147,48,647 9,223,372,036,854,775,808 to 9,223,372,036,854,775,807					
long long	8	9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	8						
float	4	3.4E +/-38	4	3.4E +/-38					
double	8	1.7E +/-308	8	1.7E +/-308					

Qualifiers (modifiers) for integer type

- signed, unsigned qualifiers can be applied to integer type
 - Signed: default. Left most bit signifies sign 0: positive 1: negative
 - Unsigned: positive. Left most bit contributes to magnitude too
 - [signed] char
 - [signed] int
 - [signed] short int
 - [signed] long int



- unsigned int
- unsigned short int
- unsigned long int

Java: no direct support for unsigned int -- always signed

unsigned int 0~2³²-1 2³² values Max: 1111111....[11111 (signed) int -2³¹~2³¹-1 2³² values Max: 0111111....11111

Qualifiers for floating points

- "long" can be used with double:
 - long double
- Thus, there are three types of floating points:
 - float /* single-precision floating point */
 - double /* double-precision floating point */
 - long double /* extended-precision floating point */
- More bits, more precise: 3.1415926535....
- printf/scanf("%f") for float, ("%lf") for double, ("%Lf") for long double
- Storage of floating point is complicated.
 - float x=4.8, float y = 6.4/2+1.6; x == y may not always true.

Qualifiers for floating points

- "long" can be used with double:
 - long double

٠

- Thus, there are three types of floating points:
 - float /* single-precision floating point */
 - double /* double-precision floating point */
 - long double /* extended-precision floating point */
- More bits, more precise: 3.1415926535....
- printf/scanf("%f") for float, ("%lf") for double, ("%Lf") for long double
- Storage of floating point is complicated.
 - float x=4.8, float y = 6.4/2+1.6; x == y may not always true.
 - No unsigned. All signed Sign Exponent Mantissa $\leftarrow 1 \text{ Bit} \rightarrow \leftarrow 8 \text{ Bits} \rightarrow \leftarrow 23 \text{ Bits} \rightarrow \leftarrow = 23$

Character Constants

- A char in C is one byte (8-bit) in size (16-bit in Java)
 - Will elaborate why 8 bits,16 bits
- A constant char is specified with single quotes:
 - Regular characters: 'A', 'C', 'z', '0', '#', '\$',...
 - char x = 'A';
- Special characters: invisible or control chars
 - New line, tab, del
 - Use escape sequence to represent

Special Characters

Escape sequence	Meaning
\n	New line
\t	Tab
\0	The null character
//	The \ character
\"	Double quote
۷'	Single quote

char c = '\t'; char c2 = '\n';

Internal Representation of characters

- characters as 1/0 bits. So they are stored as (small) integer values, interpreted according to the character set encoding (usually ASCII, 7 bits for 128 characters),
 - 'a' has encoding 97, '0' has 48, '9' has 57
- •Escape sequences are integers too
 - e.g. '\n' has10 (newline character)
 - '\t' has9 (horizontal tab)
- Special escape: '\0'
 - has encoding 0 -the null character

Internal Representation of Characters

$ \begin{array}{c c c c c c c c c c c c c c c c c c c $	<u>Dec</u>	Hx	Oct	Cha	,	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	: Hx	Oct	Html C	<u>hr</u>
220.02STX (start of text)34220.42 $\epsilon # 34; "$ 66421.02 $\epsilon \# 66; B$ 98621.42 $\epsilon \# 99; C$ 330.03ETX (end of text)35230.43 $\epsilon \# 33; \#$ 67431.03 $\epsilon \# 07; C$ 99631.43 $\epsilon \# 99; C$ 550.05ENO (enquiry)37250.45 $\epsilon \# 33; \#$ 67431.03 $\epsilon \# 07; F$ 1.06641.44 $\epsilon \# 100; C$ 660.06ACK (acknowledge)38260.46 $\epsilon \# 33; \#$ 70461.06 $\epsilon \# 102; F$ 1.05661.46 $\epsilon \# 102; F$ 77070TEL (bell)39270.47 $\epsilon \# 33; \#$ 70461.06 $\epsilon \# 102; F$ 1.05691.51 $\epsilon \# 103; G$ 880.10BS(backspace)40280.50 $\epsilon \# 412; T$ 74441.02 $\epsilon \# 103; G$ 1.05691.51 $\epsilon \# 103; G$ 10AA12B0.53 $\epsilon \# 412; T$ 74441.12 $\epsilon \# 103; G$ 1.05691.51 $\epsilon \# 103; G$ 11B0.13VT(vertical tab)422.05 $\epsilon \# 42; T$ 74441.12 $\epsilon \# 103; G$ 13D<15	0	0	000	NUL	(null)	32	20	040	 <i>€</i> #32;	Space	64	40	100	@	0	96	60	140	`	1
3 3 003 ETX (end of text)35 23 043 $6#35; #$ 67 43 103 $6#67; C$ 99 63 143 $6#99; C$ 4 4 004 EOT (end of transmission)36 24 044 $6#36; *$ 68 44 104 $6#66; D$ 100 64 144 $6#100; d$ 5 5 005 ENO (enquiry)37 25 045 $6#37; *$ 69 45 105 $6#69; E$ 101 65 145 $6#101; e$ 6 6 006 ACK (acknowledge)38 26 046 $6#38; a$ 70 46 106 $6#70; F$ 102 66 146 $6#102; f$ 7 7 007 BEL (bell)39 27 047 $6#39; '$ 71 47 107 $6#71; G$ 103 67 147 $6#103; g$ 8 8 010 BS (backspace)40 28 050 $6#40; ($ 72 48 110 $6#72; H$ 104 68 150 $6#104; h$ 9 9 011 TAB (horizontal tab)41 29 051 $6#41;)$ 73 49 111 $c#73; I$ 105 69 151 $6#105; i$ 10 A 012 LF (NL line feed, new line)42 20 052 $c#42; +$ 74 4A 112 $c#74; J$ 106 6A 152 $c#106; j$ 11 B 013 VT (vertical tab)42 20 055 $6#43; +$ 76 4C 114 $c#76; L$ 108 6C 154 $c#108; l$ 13 D 015 CR (carriage return)45 2D 055 $c#442; -$ 77 4D 115 $c#77; M$ 109 6D 155 $c#100; n$ 14 E 016 50 (shift out)46 2E 056 $c#46i; -$ 78 4E 116 $c#78; M$ 110 6E 156 $c#110; n$ 15 F 017 SI (shift in)47 2F 057 $c#471; /$ 79 4F 117 $c#79; 0$ 111 6F 157 $c#111; 0$ 16 10 020 DLE (deta link escape)48 30 060 $c#48i; 0$ 80 50 120 $c#80; P$ 112 70 160 $c#112; P$ 17 11 021 DC1 (device control 3)52 34 064 $c#52; 4$ 84 54 124 $c#84; T$ 116 74 164 $c#116; t$ 18 12 022 DC2 (device control 4)51 33 063 $c#51; 3$ 83 53 123 $c#83; S$ 115 73 163 $c#112; P$ 21 16 026 SYN (synchronous idle)53 7 067 $c#55; 7$ <td>1</td> <td>1</td> <td>001</td> <td>SOH</td> <td>(start of heading)</td> <td>33</td> <td>21</td> <td>041</td> <td>&#33;</td> <td>1</td> <td>65</td> <td>41</td> <td>101</td> <td>A</td> <td>A</td> <td>97</td> <td>61</td> <td>141</td> <td>∉#97;</td> <td>а</td>	1	1	001	SOH	(start of heading)	33	21	041	&# 33;	1	65	41	101	A	A	97	61	141	 ∉#97;	а
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	2	2	002	STX	(start of text)	34	22	042	∝# 34;	"	66	42	102	B	в	98	62	142	b	b
55005ENQ(enquiry)3725045 $6\#37; \$$ 6945105 $6\#69; E$ 10165145 $6\#101; e$ 6006ACK(acknowledge)3826046 $6\#38; \bullet$ 7046106 $4\#70; F$ 10266146 $6\#102; f$ 77070BEL(bell)3927047 $6\#39; \bullet$ 7147107 $6\#71; F$ 10367147 $6\#103; g$ 88010BS(backspace)4028050 $e\#40; ($ 7248110 $6\#72; H$ 10466150 $6\#104; h$ 99011TAB(horizontal tab)4129051 $e\#412; *$ 7349111 $6\#73; K$ 10666153 $e\#106; j$ 11B013VT(vertical tab)4220052 $e\#42; *$ 7444112 $e\#74; J$ 10666153 $e\#106; j$ 12C014FF(NP form feed, new page)4420055 $e\#45; -$ 7740115 $e\#76; E$ 10761153 $e\#107; K$ 14E016Sift ut)4522055 $e\#45; -$ 7740115 $e\#76; E$ 10865 $e\#102; n$ 15F017SI(shift ut)4525055 $e\#75; E$ 10761153 $e\#102; n$ 16100	3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	С	99	63	143	c	С
66006ACK(acknowledge)38260466#38: 670461066#70; F102661466#102; f77007BEL(bell)39270476#39; '7147103671476#103; g88010BS(backspace)40280506#40; (72481106#72; H104681506#104; h99011TAB(backspace)40280506#41;)73491116#72; H104681506#104; h10A012LF(NLline feed, new line)4220.526#42; *744A1126#74; J1066A1526#106; j11B013 <vt< td="">(verticical tab)432B0536#45; -744A1126#77; M1066A1526#106; j13D015CR(carriage return)452D0556#45; -774D1156F7; N1106E1566#100; n15F017SI(shift ut)472F0576#47; /79941116F7; N1106E1566#102; n16102DC1(device control 1)49310616#49; 181511216#63; 4#3; 011071756#112; 01711</vt<>	4	4	004	EOT	(end of transmission)	36	24	044	∝# 36;	ę.	68	44	104	D	D	100	64	144	≪#100;	d
77007BEL (bell)3927047 $\epsilon # 39; t'$ 7147107 $\epsilon \# 71; t''_{1}$ G10367147 $\epsilon \# 103; t''_{1}$ 88010BS(backspace)4028050 $\epsilon \# 410; t''_{1}$ 7248110 $\epsilon \# 72; t''_{1}$ H10468150 $\epsilon \# 104; th$ 99011TAB(horizontal tab)4129051 $\epsilon \# 412; t''_{1}$ 7444112 $\epsilon \# 73; t''_{1}$ 10666152 $\epsilon \# 105; t''_{1}$ 11B013VT(vertical tab)4220205 $\epsilon \# 44; t''_{1}$ 7444112 $\epsilon \# 73; t''_{1}$ 10666152 $\epsilon \# 106; t''_{1}$ 12C014FF(NP form feed, new page)4422053 $\epsilon \# 44; t''_{1}$ 764C114 $\epsilon \# 75; t''_{1}$ 1086C154 $\epsilon \# 106; t''_{1}$ 13D015CR(carriage return)452D055 $\epsilon \# 44; t''_{1}$ 7441115 $\epsilon \# 77; t''_{1}$ 1096D155 $\epsilon \# 100; t''_{1}$ 14E01630 (shift out)472F057 $\epsilon \# 44; t''_{1}$ 744116 $\epsilon \# 103; t''_{1}$ 116 $\epsilon \# 103; t''_{1}$ 15F017SI(shift out)472F055 $\epsilon \# 44; t''_{1}$ 74116 $\epsilon \# 103; t''_{1}$ 116 $\epsilon \# 103; t''_{1}$ 1610020 <td>5</td> <td>5</td> <td>005</td> <td>ENQ</td> <td>(enquiry)</td> <td>37</td> <td>25</td> <td>045</td> <td>∝#37;</td> <td>*</td> <td></td>	5	5	005	ENQ	(enquiry)	37	25	045	∝# 37;	*										
88010BS(backspace)4028050 $e#40$; (7248110 $e#72$; H10468150 $e#104$; h99011TAB(horizontal tab)4129051 $e#41$;)7349111 $e#73$; I1066A152 $e#106$; j10A012LF(NL line feed, new line)422A052 $e#42$; +7444112 $e#74$; J1066A152 $e#107$; k12C014FF(NP form feed, new page)442C053 $e#43$; +754B113 $e#75$; K1076B153 $e#107$; k13D015CR(carriage return)452D055 $e#45$; -774D115 $e#77$; M1096D155 $e#108$; 114E016SO(shift out)462E056 $e#46$; -774D115 $e#77$; M1096D155 $e#102$; n1610020DLE(data link escape)4830060 $e#48$; O8050120 $e#83$; S116 $e#112$; P1711021DC1(device control 1)4931061 $e#49$; 18151121 $e#63$; S1177161 $e#112$; P1711021DC1(device control 4)5234064 $e#52$; 48454124<	6	6	006	ACK	(acknowledge)	38	26	046	∝# 38;	6										
99011TAB(horizontal tab)4129051 $\epsilon \#41;$ 7349111 $\epsilon \#73;$ I10569151 $\epsilon \#105;$ 110A012LF(NL line feed, new line)422A052 $\epsilon \#42;$ 744A112 $\epsilon \#74;$ J1066A152 $\epsilon \#106;$ 111B013VT(vertical tab)432B053 $\epsilon \#42;$ 754B113 $\epsilon \#75;$ K1076B153 $\epsilon \#107;$ k12C014FF(NP form feed, new page)442C054 $\epsilon \#44;$ 76L1086C155 $\epsilon \#109;$ n14E016S0(shift out)452D055 $\epsilon \#45;$ 774D115 $\epsilon \#77;$ M1096D155 $\epsilon \#109;$ n15F017SI(shift in)472F057 $\epsilon \#47;$ 794F117 $\epsilon \#79;$ 01116F157 $\epsilon \#111;$ n1610020DLE(data link escape)4830060 $\epsilon \#49;$ 8151121 $\epsilon \#81;$ 011371161 $\epsilon \#112;$ n1106F157 $\epsilon \#111;$ n17101011deta link escape)5032062 $\epsilon \#52;$ 85551115736111;n12016 $\epsilon \#1$	7	7	007	BEL	(bell)	39	27	047	∝# 39;	1	71	47	107	G	G					
10A 012LF(NL line feed, new line)422A 052 $\epsilon #42; *$ 744A 112 $\epsilon #74; J$ 1066A 152 $\epsilon #106; j$ 11B 013VT(vertical tab)432B 053 $\epsilon #43; +$ 754B 113 $\epsilon #74; J$ 1066A 152 $\epsilon #106; j$ 12C 014FF(NP form feed, new page)442C 054 $\epsilon #44; ,$ 764C 114 $\epsilon #76; L$ 1086C 154 $\epsilon #108; I$ 13D 015CR(carriage return)452D 055 $\epsilon #44; -$ 764C 114 $\epsilon #77; M$ 1096D 155 $\epsilon #109; II$ 14E 016SO(shift out)462E 056 $\epsilon #46; .$ 784E 116 $\epsilon #78; N$ 1106E 156 $\epsilon #110; II$ 15F 017SI(shift in)472F 057 $\epsilon #47; /$ 794F 117 $\epsilon #79; O$ 1116F 157 $\epsilon #111; O$ 1610020DLE(device control 1)4931061 $\epsilon #49; I$ 8050120 $\epsilon #81; O$ 11270160 $\epsilon #112; P$ 171022DC2(device control 3)5133063 $\epsilon #51; 3$ 8353123 $\epsilon #83; S$ 11573163 $\epsilon #114; r$ 1913023DC3(device control 4)5234066 $\epsilon #51; 6$ 8555555511674164 $\epsilon #116; T$ 2115025MAK(negative acknow	8				(backspace)															
11B013VT(vertical tab)432B053 $6\#43$; +754B113 $6\#75$; K1076B153 $6\#107$; k12C014FF(NP form feed, new page)442C054 $6\#44$; ,764C114 $6\#75$; L1086C154 $6\#107$; k13D015CR(carriage return)452D055 $6\#45$; .774D115 $6\#77$; M1096D155 $6\#109$; n14E016S0(shift out)472F05 $6\#46$; .784E116 $6\#77$; M1096D155 $6\#109$; n15F017SI(shift out)472F057 $6\#477$; /794F116 $6\#78$; N1106E156 $6\#109$; n1610020DLE(data link escape)47755576 $6\#77$; N1096D155 $6\#112$; p1711021DC1(device control 1)4931061 $6\#49$; 18151121 $6\#83$; 2R11472162 $6\#114$; r1913023DC3(device control 4)5234066 $6\#52$; 48454124 $6\#64$; T11674164 $6\#113$; r2115025NAK(negative acknowledge)5335065 $6\#53$; 58555125	9				(horizontal tab)															
12C014FF(NP form feed, new page)442C054 $\epsilon #44;$ 764C114 $\epsilon #76;$ 11086C154 $\epsilon #108;$ 113D015CR(carriage return)452D055 $\epsilon #45;$ 774D115 $\epsilon #77;$ M1096D155 $\epsilon #109;$ m14E016SO(shift out)462E056 $\epsilon #46;$ 784E116 $\epsilon #78;$ N1106E156 $\epsilon #111;$ 01610020DLE(data link escape)483O606 $\epsilon #49;$ 18151121 $\epsilon #81;$ 01116F157 $\epsilon #111;$ 0171010(device control 1)4931061 $\epsilon #49;$ 18151121 $\epsilon #81;$ 11371161 $\epsilon #113;$ q1812022DC2(device control 3)5133063 $\epsilon #51;$ 8353123 $\epsilon #83;$ 11573163 $\epsilon #114;$ r1913023DC3(device control 4)5234064 $\epsilon #52;$ 8454124 $\epsilon #84;$ 11674164 $\epsilon #116;$ t2115025NAK(negative acknowledge)5335065 $\epsilon #53;$ 8555125 $\epsilon #85;$ 11775165 $\epsilon #117;$ u11876	10				(NL line feed, new line)															
13D015CR(carriage return)452D055 $6\#45;$ 774D115 $6\#77;$ M1096D155 $6\#109;$ m14E016SO(shift out)462E056 $6\#46;$ 784E116 $6\#77;$ M1096D155 $6\#109;$ m15F017SI(shift in)472F057 $6\#47;$ 794F117 $6\#77;$ M1106E156 $6\#10;$ n1610020DLE(data link escape)4830060 $6\#49;$ 18050120 $6\#80;$ P11270160 $6\#112;$ P1711021DC1(device control 1)4931061 $6\#49;$ 18151121 $6\#81;$ 011371161 $6\#112;$ P1913023DC3(device control 4)5133063 $6\#51;$ 8353123 $6\#83;$ S11472162 $6\#114;$ r2115025NAK(negative acknowledge)5335065 $6\#53;$ 8555125 $6\#85;$ U11775165 $6\#117;$ U2216026SYN(synchronous idle)5436066 $6\#54;$ 6858130 $6\#86;$ V1187666 $6\#148;$ V<	11				(vertical tab)					+										
14E01630(shift out)462E056 $\& 466$;784E1106E156 $\& #110$; n15F017SI(shift in)472F057 $\& 477$;794F117 $\& 479$; 01116F157 $\& #111$; 01610020DLE(data link escape)4830060 $\& 448$; 08050120 $\& 805$; P11270160 $\& #112$; p1711021DC1(device control 1)4931061 $\& 492$; 18151121 $\& 818$; Q11371161 $\& #113$; Q1812022DC2(device control 3)5032062 $\& 450$; 28353123 $\& 483$; S11674164 $\& #116$; t1913023DC3(device control 4)5234066 $\& 455$; 48454124 $\& 484$; T11674164 $\& 4116$; t2115025NAK(negative acknowledge)5436066 $\& 455$; 58555125 $\& 865$; 126 $\& 865$; 126 $\& 865$; 127 $\& 87$ 11774164 $\& 4116$; t2216026SYN(synchronous idle)5436066 $\& 55$; 78757127 $\& 87$ 11977167 $\& 4112$; y2418030CAN(cancel)5537	12				(NP form feed, new page)					10										
15F017SI(shift in)472F057 $\% 477$;794F117 $\% 79$;01116F157 $\% \# 111$;01610020DLE(data link escape)4830060 $\% 448$;08050120 $6 \# 80$;P11270160 $6 \# 112$;P1711021DC1(device control 1)4931061 $6 \# 49$;18151121 $6 \# 81$;011371161 $6 \# 113$;q1812022DC2(device control 2)5032062 $6 \# 50$;28252122 $6 \# 32$;R11472162 $6 \# 114$;r1913023DC3(device control 3)5133063 $6 \# 51$;38353123 $6 \# 32$;R11472162 $6 \# 114$;r2115025NAK(negative acknowledge)5335065 $6 \# 55$;8555125 $6 \# 65$;U11775165 $6 \# 117$;u2115025NAK(negative acknowledge)5537067 $6 \# 55$;8555125 $6 \# 65$;U11775165 $6 \# 117$;u2317027ETB(end of trans. block)5537067 $6 \# 55$;8555127 $6 \# 67$; <td>13</td> <td></td> <td>_</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	13											_								
16 10 020 DLE (data link escape) 48 30 060 \$\$48; 0 80 50 120 \$\$\$80; P 112 70 160 \$																				
17 11 021 DC1 (device control 1)49 31 061 $\&$ #49; 181 51 121 $\&$ #81; Q113 71 161 $\&$ #113; Q18 12 022 DC2 (device control 2)50 32 062 $\&$ #50; 282 52 122 $\&$ #82; R114 72 162 $\&$ #114; r19 13 023 DC3 (device control 3)51 33 063 $\&$ #51; 383 53 123 $\&$ #83; S115 73 163 $\&$ #115; S20 14 024 DC4 (device control 4)52 34 064 $\&$ #52; 484 54 124 $\&$ #84; T116 74 164 $\&$ #116; t21 15 025 NAK (negative acknowledge)53 35 065 $\&$ #53; 584 54 124 $\&$ #84; T116 74 164 $\&$ #116; t22 16 026 SYN (synchronous idle)54 36 066 $\&$ #54; 686 56 126 $\&$ #86; V118 76 166 $\&$ #118; V23 17 027 ETB (end of trans. block)55 37 067 $\&$ #55; 787 57 127 $\&$ #87; W119 77 167 $\&$ #119; W24 18 030 CAN (cancel)56 38 070 $\&$ #56; 888 58 130 $\&$ #88; X120 78 170 $\&$ #120; X25 19 031 EM (end of medium)57 39 071 $\&$ 57; 989 59 131 $\&$ #89; Y121 79 171 $\&$ #121; Y26 1A 032 SUB (substitute)58 3A 072 $\&$ #58; :90 5A 132 $\&$ 90; Z122 7A 172 $\&$ #122; Z27 1B 033 ESC (escape)59 3B 073 $\&$ 459; :91 5B 133 $\&$ 91; [123 7B 173 $\&$ #123; {28 1C 034 FS (file separator)61 3D 075 $\&$ #61; =93 5D 135 $\&$ 93;]125 7D 175 $\&$ #124; !29 1D 035 GS (group separator)61 3D 075 $\&$ #61; =93 5D 135 $\&$ 99;]126 7E 176 $\&$ #126; ~31 1F 037 US (unit separator)63 3F 077 $\&$ #63; ?95 5F 137 $\&$ 99; _127 7F 177 $\&$ #127; DEL					· ·															
1812022DC2(device control 2)5032062 $\&\#50; 2$ 825211472162 $\&\#114; r$ 1913023DC3(device control 3)5133063 $\&\#51; 3$ 835311573163 $\&\#115; s$ 2014024DC4(device control 4)5234064 $\&\#52; 4$ 8454124 $\&\#84; T$ 11674164 $\&\#116; t$ 2115025NAK(negative acknowledge)5335065 $\&\#53; 5$ 8555125 $\&\#85; U$ 11775165 $\&\#117; u$ 2216026SYN(synchronous idle)5436066 $\&\#54; 6$ 8656126 $\&\#86; V$ 11876166 $\&\#118; V$ 2317027ETB(end of trans. block)5537067 $\&\#55; 7$ 8757127 $\&\#87; W$ 11977167 $\&\#119; W$ 2418030CAN(cancel)5638070 $\&\#55; 7$ 8757127 $\&\#87; W$ 11977167 $\&\#120; X$ 2519031EM(end of medium)5739071 $\&\#57; 9$ 8959131 $\&\#89; Y$ 12179171 $\&\#120; X$ 261A032SUB(substitute)583A072 $\&\#59; 2$ 9158133 $\&\#99; Y$ 122 </td <td></td>																				
1913023DC3(device control 3)51330633 38353123S S11573163s S2014024DC4(device control 4)52340644 48454124T T11674164t t2115025NAK(negative acknowledge)53350655 58555125U U11775165u u2216026SYN(synchronous idle)54360667 78757127W W11977167w W2418030CAN(cancel)56380707 78757127W W11977167p Y261A032SUB(substitute)583A072: :905A132Z Z1227A172z Z271B033ESC(escape)593B073; :915B133[[1237B173{ {281C034FS(file separator)603C074< <															_					
2014024DC4 (device control 4)5234064 4 48454124 T T11674164 t t2115025NAK (negative acknowledge)5335065 5 58555125 U U11775165 u u2216026SYN (synchronous idle)5436066 6 68655125 U U11775165 u u2317027ETB (end of trans. block)5537067 7 7865611977167 w W2418030CAN (cancel)5638070 8 88858130 X X12078170 x X2519031EM (end of medium)5739071 9 98959131 Y Y12179171 y Y261A032SUB (substitute)583A072 : :905A132 Z Z1227A172 z Z271B033ESC (escape)593B073 ; ;915B133 [[1237B173 { {281C034FS(file separator)603C074 < <																				
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; { 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																				
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; <																				
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; <																				
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; <																				
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; <																				
26 1A 032 SUB (substitute) 58 3A 072 :: 90 5A 132 Z Z 122 7A 172 z Z 27 1B 033 ESC (escape) 59 3B 073 ;; 91 5B 133 [[123 7B 173 { { 28 1C 034 FS (file separator) 60 3C 074 < <																				
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; <																				
28 1C 034 FS (file separator) 60 3C 074 < <																				
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										-					-					
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																				
31 1F 037 US (unit separator) 63 3F 077 & #63; 2 95 5F 137 & #95; 127 7F 177 & #127; DEL															_					
	31	lF	037	US	(unit separator)	63	ЗF	077	?	2	95	5F	137	«#95;	_	127				DEL

Source: www.LookupTables.com

Characters

- **chars** are treated in C (and Java) as small integers, **char** variables and constants are identical to **int** in arithmetic expressions:
 - char c is converted to its encoding (index in the character set table)
- char aChar= 'E'; // encoding 69
 'E' + 8 //expression with value 69+8 = 77
 'E' + 'B' //expression with value 69+66 = 135
 'E' -'B' //expression with value 69-66 = 3
- Same for other expressions. In relational/logical expression, characters can be compared directly, comparing indexes/encodings
- aChar== EOF //index == -1?
- aChar== 'H' //index == 72?
- aChar== '\n' // index == 10?
- aChar< 'H' //69 < 72?

Characters

• Since **chars** are just small integers, **char** variables and constants are identical to **int** in arithmetic expressions. Some programming idioms that take advantage of this:

printf("numerical value is %d\n", c-'0')



same in Java

}

Example

```
#include<stdio.h>
/*copying input to output with
converting upper-case letters to lower-case */
main() {
   int c; int outC;
   c = getchar();
   while (c != EOF)
    Ł
       if (c >= 'A' && c <= 'Z') /* 65 \sim 90 upper case letter*/
           outC = c + ('a' - 'A'); /* = c + 'b' - 'B' */
       else
                                      /* = c + ('c' - 'C') */
          outC = c;
                                      /* = c + 'z' - 'Z' */
                                      /* = tolower(c) */
       putchar(outC);
                                                      c + 32 works but
       c = getchar(); //| read again
                                                      not good for portability.
    }
                                                      Avoid that!
```