Agenda

Tools for C programming: C Tutor, GDB, and Valgrind

Binary and data representations

Unsigned integers

Signed integers

Addition and subtraction

Overflow

Signed extension

GDB/Valgrind Demo: bigfish.c

```
// allocate space for two int arrays
bigfish = (int *)malloc(sizeof(int)*10);
littlefish = (int *)malloc(sizeof(int)*10);
for (i=0; i < 10; i++) {
    bigfish[i] = 10+i;
    littlefish[i] = i;
}
print_array(bigfish,10, "bigfish");
print_array(littlefish,10, "littlefish");
for (i=0; i < 13; i++) {
    bigfish[i] = 66+i;
}
printf("\nafter loop:\n");
print_array(bigfish,10, "bigfish");
print_array(littlefish,10, "littlefish");</pre>
```

GDB/Valgrind Demo: bigfish.c

```
// allocate space for two int arrays
bigfish = (int *)malloc(sizeof(int)*10);
littlefish = (int *)malloc(sizeof(int)*10);
for (i=0; i < 10; i++) {
    bigfish[i] = 10+i;
    littlefish[i] = i;
}
print_array(bigfish,10, "bigfish");
print_array(littlefish,10, "littlefish");
for (i=0; i < 13; i++) {
    bigfish[i] = 66+i;
}
printf("\nafter loop:\n");
print_array(bigfish,10, "bigfish");
print_array(littlefish,10, "littlefish");</pre>
```

```
bigfish array:
10 11 12 13 14 15 16 17 18 19
littlefish array:
0 1 2 3 4 5 6 7 8 9

after loop:
bigfish array:
66 67 68 69 70 71 72 73 74 75
littlefish array:
78 1 2 3 4 5 6 7 8 9
Segmentation fault (core dumped)
```

GDB/Valgrind Demo: badprog.c

```
int findAndReturnMax(int *array1, int len, int max) {
    int i;
    if (!array1 || (len <=0) ) {
        return -1;
    }
    max = array1[0];
    for (i=1; i <= len; i++) {
        if (max < array1[i]) {
            max = array1[i];
        }
    }
    return 0;
}</pre>
```

```
int main(int argc, char *argv[]) {
  int arr[5] = { 17, 21, 44, 2, 60 };
  int max = arr[0];

if ( findAndReturnMax(arr, 5, max) != 0 ) {
    printf("strange error\n");
    exit(1);
  }
  printf("max value in the array is %d\n", max);
  return 0;
}
```

What is the output of this program supposed to be?

Binary and Data Representation

Data is stored as **binary** signals e.g. they can either be **on** or **off**

Each signal corresponds to a single bit

All data can be represented with bits more complicated data -> needs more bits

Binary and data representation

Smallest unit of addressable memory is a byte

Memory address 0: 01010101 Memory address 1: 10101010 Memory address 2: 00001111

A byte is 8 bits

A **word** is the default size of memory that the hardware moves around either 32 bits or 64 bits

Why define variables? Why have types?

Variable types in C

• Different Types have different number of bytes:

```
1 byte: char, unsigned char (no negative values)
2 bytes: short, unsigned short
4 bytes: int, unsigned int, float
8 bytes: long long, unsigned long long, double
4 or 8 bytes: long, unsigned long
```

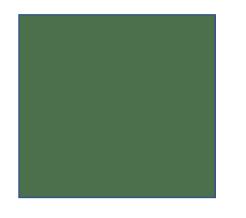
NOTE: On a 64 bit machine, pointers are $\frac{2}{2}$ bytes

Example: Memory can be interpreted in different ways depending on the context

Consider the data

0b010011000110111101001100 (0x4C6F4C)

The above bits can mean any of the following



LOL

5,009,228

Example: ASCII

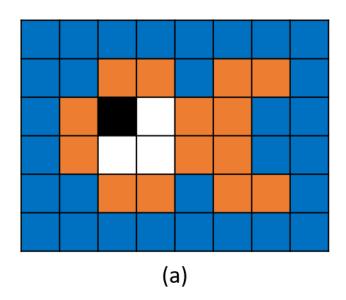
American Standard Code for Information Interchange

0 0 000 NUL (null) 1 1 001 SOH (start of heading) 2 2 002 STX (start of text) 3 2 1 041 (#333; ! 65 41 101 (#55; A 97 61 141 (#97; A 97 41 141 (#97; A 97 41 141 (#97; A 97 41 141 (#97; A 141 141 (#97; A 141 141	Dec Hx Oct Char	Dec Hx Oct Html Chr	Dec Hx Oct Html Chr Dec Hx Oct Html Chr
1 1 001 SOH (start of heading) 2 2 002 SYK (start of text) 3 3 003 ETX (end of text) 3 3 20 042 c#34;" 6 6 42 102 c#66; B 9 8 62 142 c#98; b 3 3 003 ETX (end of text) 4 4 004 EOT (end of transmission) 5 5 005 ENQ (enquiry) 3 7 25 045 c#37; c 6 6 006 ACK (acknowledge) 3 8 26 046 c#38; c 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 1 B 013 VT (vertical tab) 1 B 013 VT (vertical tab) 1 B 013 VT (vertical tab) 1 C 014 FF (NL line feed, new page) 1 S F 017 SI (shift in) 1 S F 017 SI (shift in) 1 S F 017 SI (shift in) 1 C 10 20 DLE (data link escape) 1 T 1 0 10 20 DLE (device control 1) 1 B 10 302 DC2 (device control 1) 1 B 10 303 DC3 (device control 3) 2 1 1 0 22 DC2 (device control 3) 2 1 1 0 23 DC3 (device control 4) 2 1 1 0 23 DC3 (device control 4) 2 1 1 0 23 DC3 (device control 4) 2 1 1 1 0 23 DC3 (device control 4) 2 1 1 1 0 23 DC3 (device control 4) 2 1 1 1 0 23 DC3 (device control 4) 2 1 1 1 0 23 DC3 (device control 4) 2 1 1 1 0 23 DC3 (device control 4) 2 1 1 1 0 23 DC3 (device control 4) 2 1 1 1 0 24 DC4 (device control 4) 2 1 1 1 0 25 DAK (negative acknowledge) 3 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 000 NUL (null)	32 20 040 Space	64 40 100 6#64; 0 96 60 140 6#96;
2 2 002 STX (start of text) 3 3 003 STX (end of text) 4 4 004 EOT (end of transmission) 5 5 005 ENO (enquiry) 6 6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NF form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 1) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 3) 20 14 024 DC4 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (engative acknowledge) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 3 3 003 EN (end of text) 36 20 044 6#36; 6 44 104 6#68; D 70 46 106 6#00; E 70 46 106 6#70; F 70 46 106 6	· · ·		
3 3 003 ETX (end of text) 4 4 004 EOT (end of transmission) 5 5 005 ENQ (enquiry) 6 6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (ML line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 S0 (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 18 12 022 DC2 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 23 17 027 ETB (end of transmission) 35 23 043 £#35; # 66 44 103 6#67; C			
5 5 005 ENQ (enquiry) 6 6 6 006 ACK (acknowledge) 7 7 007 BEL (bell) 8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 1) 19 13 023 DC3 (device control 2) 20 14 Q24 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 25 19 031 EM (end of medium) 25 19 10 303 CS (group separator) 26 12 16 026 RS (record separator) 27 16 03 6 RS (record separator) 28 16 70 46 6 393; ** 40 28 050 4#40; (72 48 110 6#72; H) 40 28 050 6#40; (72 48 110 6#72; H) 41 29 051 6#41;) 73 49 111 6#73; I 105 69 151 6#105; 1 41 29 051 6#41;) 73 49 111 6#73; I 105 69 151 6#105; 1 41 29 051 6#41;) 74 44 112 6#74; J 106 6A 152 6#105; 1 41 29 051 6#41;) 74 44 112 6#75; K 107 6B 153 6#107; k 42 20 052 6#46; . 77 4D 115 6#77; M 109 6D 155 6#109; m 44 20 054 6#46; . 77 4D 115 6#77; M 109 6D 155 6#109; m 45 2D 055 6#46; . 78 447; / 79 4F 117 6#79; O 111 6F 157 6#111; O 110 6E 166 6#110; n 47 2F 057 6#47; / 79 4F 117 6#79; O 111 6F 157 6#111; O 111 6E 156 6#110; n 47 2F 057 6#45; O 32 062 6#50; O 32 062 6#50; O 32 062 6#50; O 32 062 6#50; O 32 063 6#51; O 33 063 6#51; O 34 064 6#60; O 34 064 064; O 34 064 064 064 064 064 064 064 064 064 06	·		67 43 103 4#67; C 99 63 143 4#99; C
6 6 006 ACK (acknowledge) 38 26 046 6#38; 6 70 46 106 6#70; F 102 66 146 6#102; f 7 7 007 EEL (bell) 39 27 047 6#39; 1 71 47 107 6#71; G 103 67 147 6#103; g 8 8 010 BS (backspace) 40 28 050 6#40; (7 2 48 110 6#72; H 104 68 150 6#104; h 129 051 6#41;) 73 49 111 6#73; I 105 69 151 6#105; i 11 B 013 VT (vertical tab) 42 2A 052 6#42; 7 74 4A 112 6#74; J 106 6A 152 6#106; j 11 B 013 VT (vertical tab) 42 C 054 6#44; 7 75 4B 113 6#75; K 107 6B 153 6#107; k 12 C 014 FF (NP form feed, new page) 14 E 016 50 (shift out) 42 C 056 6#46; . 44 12 0 055 6#45; - 77 4D 115 6#77; M 109 6D 155 6#109; m 14 E 016 50 (shift out) 47 2F 057 6#47; 7 79 4F 117 6#79; 0 111 6F 157 6#111; 0 10 020 DLE (data link escape) 48 30 060 6#48; 0 80 50 120 6#80; P 112 70 160 6#112; P 113 022 DC2 (device control 1) 49 31 061 6#50; 2 82 51 122 6#80; P 112 70 160 6#112; P 113 023 DC3 (device control 3) 51 33 063 6#51; 3 83 53 123 6#83; S 115 73 163 6#117; u 12 15 025 NAK (negative acknowledge) 54 36 066 6#54; 6 86 55 126 6#86; V 117 75 165 6#117; u 12 161 030 CAN (cancel) 57 39 071 6#57; 9 87 57 127 6#87; U 119 77 167 6#119; W 24 18 030 CAN (cancel) 58 3A 072 6#58; 9 131 6#89; Y 12 79 171 6#121; Y 29 1D 035 GS (group separator) 63 3F 077 6#63; 2 95 5F 137 6#95; 1 127 7F 177 6#127; DEL	4 4 004 EOT (end of transmission)	36 24 044 \$ \$	68 44 104 6#68; D 100 64 144 6#100; d
7 7 007 BEL (bell) 39 27 047 6#39;	5 5 005 ENQ (enquiry)	37 25 045 6#37; %	
8 8 010 BS (backspace) 9 9 011 TAB (horizontal tab) 10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 10 036 CB (group separator) 27 18 037 US (unit separator) 28 16 036 CB (carciage return) 40 28 050 6#40; (72 48 110 6#72; H 105 69 151 6#104; h 105 6#115; h	6 6 006 ACK (acknowledge)	38 26 046 & &	
9 9 011 TAB (horizontal tab) 41 29 051 6#41;) 73 49 111 6#73; I 105 69 151 6#105; i 10 A 012 LF (NL line feed, new line) 42 2A 052 6#42; * 74 4A 112 6#74; I 106 6A 152 6#106; j 118 013 VT (vertical tab) 42 2B 053 6#43; + 75 4B 113 6#75; K 107 6B 153 6#107; k 12 C 014 FF (NP form feed, new page) 44 2C 054 6#44; , 76 4C 114 6#76; L 108 6C 154 6#108; l 13 D 015 CR (carriage return) 45 2D 055 6#45; - 77 4D 115 6#77; M 109 6D 155 6#109; m 14 E 016 SO (shift out) 46 2E 056 6#46; . 78 4E 116 6#78; N 110 6E 156 6#110; n 15 F 017 SI (shift in) 47 2F 057 6#47; / 9 4F 117 6#79; O 111 6F 157 6#111; O 11 021 DC1 (device control 1) 49 31 061 6#49; l 81 51 121 6#81; O 113 71 161 6#113; O 113 71 161 6#113; O 114 024 DC4 (device control 3) 51 33 063 6#51; 3 83 53 123 6#38; S 115 73 163 6#115; S 23 40 64 6#52; 4 46 6#55; 7 85 55 125 6#85; U 117 75 165 6#117; U 121 DC1 (carriage return) 49 31 066 6#54; 6 86 56 126 6#86; V 118 76 166 6#118; V 117 75 165 6#117; U 118 030 CAN (cancel) 57 39 071 6#55; 9 89 59 131 6#89; V 121 79 171 6#121; V 121 033 ESC (escape) 59 3B 073 6#59; 9 91 5B 133 6#99; D 124 70 6#122; D 125 7B 175 6#122; Z 180 036 RS (group separator) 60 3F 077 6#66; 9 95 5F 137 6#95; 1 127 7F 177 6#127; DEL	7 7 007 BEL (bell)	39 27 047 6#39; '	
10 A 012 LF (NL line feed, new line) 11 B 013 VT (vertical tab) 12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 43 2B 053 6#42; * 43 2B 053 6#43; + 44 2C 054 6#44; , 45 2D 055 6#44; , 76 4C 114 6#76; L 108 6C 154 6#108; l 107 4B 113 6#75; K 109 6B 153 6#107; k 107 4B 113 6#76; L 108 6C 154 6#108; l 108 6C 154 6#108; l 109 6D 155 6#109; m 110 6E 156 6#110; n 109 6D 155 6#109; m 109 6D 156 6#109; m 110 6E 156 6#110; m 111 6F 157 6#111; o 111 6F 157 6#11	8 8 010 BS (backspace)	40 28 050 6#40; (
11 B 013 VT (vertical tab) 12 C 014 FF (NF form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 23 17 027 ETB (end of medium) 25 18 033 ESC (escape) 26 18 032 SUB (substitute) 27 18 033 ESC (escape) 28 16 036 RS (group separator) 30 12 037 US (unit separator) 43 2B 053 4#43; + 44 2C 054 4#44; , 76 4C 114 4#76; L 77 4D 115 6#77; M 108 6C 154 6#108; l 77 4C 114 6#76; L 108 6C 154 6#108; l 77 4C 114 6#76; L 77 4D 115 6#77; M 110 6E 156 6#101; n 78 4E 116 6#78; N 111 6F 157 6#110; n 78 4E 116 6#78; N 110 6E 156 6#101; n 78 4E 116 6#78; N 111 6F 157 6#110; n 78 4E 116 6#78; N 110 6E 156 6#101; n 78 4E 116 6#78; N 111 6F 157 6#110; n 78 4E 116 6#78; N 110 6E 156 6#101; n 78 4E 116 6#78; N 111 75 16 6#112; p 80 50 120 6#80; P 111 7 7 167 6#112; p 81 51 33 063 6#51; 3 83 53 123 6#80; P 83 51 123 6#80; P 84 52 122 6#82; E 84 54 124 6#84;	9 9 011 TAB (horizontal tab)	41 29 051 6#41;)	
12 C 014 FF (NP form feed, new page) 13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 SWK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 10 032 GS (group separator) 27 18 033 GS (group separator) 30 12 03 RS (record separator) 31 1F 037 US (unit separator) 44 2C 054 c#44; 45 2D 055 c#44; 46 2E 056 c#445; 47 4D 115 c#77; M 109 6D 155 c#109; m 109 6D 155 c#109; m 110 6E 156 c#110; n 17 4D 115 c#77; M 109 6D 155 c#109; m 110 6E 156 c#110; n 17 4D 115 c#77; M 109 6D 155 c#109; m 110 6E 156 c#110; n 17 4D 115 c#77; M 109 6D 155 c#109; m 110 6E 156 c#110; n 111 6F 157 c#111; o 112 c#61; O 111 6F 157 c#111; o 112 c#61; O 113 07 08 (#48; O 110 6E 156 c#110; n 110 6E 156 c#111; o 110 6E 156 c#111; o 110 6E 156 c#110; n 110 6E 156 c#110; n 110 6E 156 c#110; n 110 6E 156 c#111; o 110 6E 156 c#111; o 110 6E 156 c#111; o 110 6E 156 c#110; n 110 6E 156 c#111; o 110 6E 156 c#110; n 111 6F 157 c#111; o 110 6E 156 c#110; n 111 6F 157 c#111; o 110 6E 156 c#111; o 110 6E 156 c#110; n 111 6F 157 c#111; o 110 6E 156 c#110; n 111 6F 157 c#111; o 110 6E 156 c#111; o 111 6F 157 c#112; o 111 6F 157 c#112; o 111 6F 157 c#112; o 111 6F 157 c#127; o 111 6F 157 c#112; o 111 6F 157 c#112; o 111 6F 157 c#127; o 111 6F 157 c#112; o 111	10 A 012 LF (NL line feed, new line)	42 2A 052 * *	
13 D 015 CR (carriage return) 14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 10 07 68 (group separator) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 45 2D 055 6#45; - 46 2E 056 6#46; - 47 4D 115 6#77; M 109 6D 155 6#109; m 48 46 2E 056 6#46; - 48 45; - 48 45 116 6#78; N 110 6E 156 6#110; n 79 4F 117 6#79; O 111 6F 157 6#111; o 80 50 120 6#80; P 112 70 160 6#112; p 80 50 120 6#80; P 112 70 160 6#112; p 81 51 121 6#81; O 113 71 161 6#113; q 81 51 121 6#81; O 113 71 161 6#113; q 81 51 121 6#81; O 113 71 161 6#113; q 82 52 122 6#82; R 114 72 162 6#114; r 83 53 123 6#83; S 115 73 163 6#115; S 84 54 124 6#84; T 116 74 164 6#116; t 85 55 125 6#85; U 117 75 165 6#117; u 86 56 126 6#86; V 118 74 164 6#116; t 87 77 4D 115 6#77; M 109 6D 155 6#109; m 87 84E 116 6#78; N 110 6E 156 6#111; o 111 6F 157 6#111; o 112 70 160 6#112; p 112 70 160 6#112; p 113 71 161 6#113; q 113 71 161 6#113; q 115 73 163 6#113; q 115 73 163 6#114; r 116 74 164 6#113; q 117 75 165 6#117; u 118 77 167 6#114; u 119 77 167 6#119; u 119	11 B 013 VT (vertical tab)		10 10 10 10 10 10 10 10 10 10 10 10 10 1
14 E 016 SO (shift out) 15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 38 103 CF (group separator) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 46 2E 056 . . 78 4E 116 N N 79 4F 117 O O 111 6F 157 o O 112 6F 157 o O 113 050 o O 113 050 P P 112 70 160 p P 113 03 E 15 025 NAK (epatabox control 1) 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1 113 061 1 1	12 C 014 FF (NP form feed, new page)		
15 F 017 SI (shift in) 16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 25 18 032 SUB (substitute) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 10 074 c#60; 29 1D 035 GS (group separator) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 47 2F 057 c#47; / 48 30 060 c#48; 0 49 31 061 c#49; 1 48 30 060 c#51; 3 48 51 121 c#81; 0 47 116 c#111; 0 113 71 161 c#111; 0 114 72 162 c#1114; r 116 72 162 c#114; r 116 72 162 c#12; r 117 79 171 c#121; r 118 70 172 c#12 c#12; r 119 70 172 c#12	13 D 015 CR (carriage return)		
16 10 020 DLE (data link escape) 17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 38 03 060 6#48; 0 49 31 061 6#49; 1 50 32 062 6#50; 2 62 36 076 6#55; 3 50 36 063 6#51; 3 50 36 063 6#51; 3 50 36 063 6#51; 3 50 36 064 6#52; 4 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#62; 7 50 37 067 6#55; 7 50 37 067 6#55; 7 50 37 067 6#68; 7 50 38 55 125 6#85; U 51 37 067 6#68; T 51 30 063 6#51; 3 50 36 063 6#51; 3 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50 36 066 6#54; 6 50	14 E 016 SO (shift out)	46 2E 056 . .	
17 11 021 DC1 (device control 1) 18 12 022 DC2 (device control 2) 19 13 023 DC3 (device control 3) 20 14 024 DC4 (device control 4) 21 15 025 NAK (negative acknowledge) 22 16 026 SYN (synchronous idle) 23 17 027 ETB (end of trans. block) 24 18 030 CAN (cancel) 25 19 031 EM (end of medium) 26 1A 032 SUB (substitute) 27 1B 033 ESC (escape) 28 1C 034 FS (file separator) 30 1E 036 RS (record separator) 31 1F 037 US (unit separator) 49 31 061 1 1 50 32 062 2 2 50 32 063 3 3 50 35 063 3 3 50 35 063 = 2 51 33 063 3 3 52 34 064 4 4 54 54 124 T T 56 74 164 q Q 57 3 163 s S 57 3 163 Y V 57 3 163 t 57 3 163 s S 57 3 163 s S 57 3 163 t 57 3 163 s S 57 3 163 Y V 57 3 163 t 57 3 163 s S 57 3 163 S 57 3 163 t 57 3 163 s S 57 3 163 t 57 3 163 s S 57 3 163 S 57 3 163 Y V 57 3 163 t 57 3 163 s S 57 3 163 t 57 3 163 s S 57 3 163 S 57 3 163 t 57 3 16	,		
18 12 022 DC2 (device control 2)			
19 13 023 DC3 (device control 3)			
20 14 024 DC4 (device control 4) 52 34 064 4 4 84 54 124 T T 116 74 164 t t 21 15 025 NAK (negative acknowledge) 53 35 065 5 5 85 55 125 U U 117 75 165 u u 22 16 026 SYN (synchronous idle) 54 36 066 6 6 86 56 126 V V 118 76 166 v V 23 17 027 ETB (end of trans. block) 55 37 067 7 7 87 57 127 W W 119 77 167 w W 24 18 030 CAN (cancel) 56 38 070 8 8 88 58 130 X X 120 78 170 x X 25 19 031 EM (end of medium) 57 39 071 9 9 89 59 131 Y Y 121 79 171 y Y 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 < < 92 5C 134 \ \ 124 7C 174 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ 31 1F 037 US (unit separator) 63 3F 077 ? ? 95 5F 137 _ _ 127 7F 177  DEL			I I
21 15 025 NAK (negative acknowledge)			
22 16 026 SYN (synchronous idle)			
23 17 027 ETB (end of trans. block) 55 37 067 7 7 87 57 127 W W 119 77 167 w W 24 18 030 CAN (cancel) 56 38 070 8 8 88 58 130 X X 120 78 170 x X 25 19 031 EM (end of medium) 57 39 071 9 9 89 59 131 Y Y 121 79 171 y Y 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 < < 92 5C 134 \ \ 124 7C 174 \ 129 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ \ 126 7E 176 ~ \ 28 1F 037 US (unit separator) 63 3F 077 ? 2 95 5F 137 _ \ 127 7F 177  DEL			
24 18 030 CAN (cancel) 56 38 070 8 8 88 58 130 X X 120 78 170 x X 25 19 031 EM (end of medium) 57 39 071 9 9 89 59 131 Y Y 121 79 171 y Y 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 < < 92 5C 134 \ \ 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ \ 126 7E 176 ~ \ 29 3D 135] \ 20 127 7F 177  DEL			
25 19 031 EM (end of medium) 57 39 071 6#57; 9 89 59 131 6#89; Y 121 79 171 6#121; Y 26 1A 032 SUB (substitute) 58 3A 072 6#58; 90 5A 132 6#90; Z 122 7A 172 6#122; Z 2 7 1B 033 ESC (escape) 59 3B 073 6#59; 91 5B 133 6#91; [123 7B 173 6#123; { 28 1C 034 FS (file separator) 60 3C 074 6#60; < 92 5C 134 6#92; \ 124 7C 174 6#124; 29 1D 035 GS (group separator) 61 3D 075 6#61; = 93 5D 135 6#93;] 125 7D 175 6#125; } 30 1E 036 RS (record separator) 62 3E 076 6#62; > 94 5E 136 6#94; \ 126 7E 176 6#126; \ 23 1F 037 US (unit separator) 63 3F 077 6#63; 2 95 5F 137 6#95; _ 127 7F 177 6#127; DEL			
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 < < 92 5C 134 \ \ 124 7C 174 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ \ 126 7E 176 ~ \ 23 1F 037 US (unit separator) 63 3F 077 ? 2 95 5F 137 _ _ 127 7F 177  DEL			
27 1B 033 ESC (escape) 59 3B 073 ; 91 5B 133 [[123 7B 173 { { 28 1C 034 FS (file separator) 60 3C 074 < < 92 5C 134 \ \ 124 7C 174 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ 31 1F 037 US (unit separator) 63 3F 077 ? ? 95 5F 137 _ _ 127 7F 177  DEL			
28 1C 034 FS (file separator) 60 3C 074 < < 92 5C 134 \ \ 124 7C 174 29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ 31 1F 037 US (unit separator) 63 3F 077 ? ? 95 5F 137 _ _ 127 7F 177  DEL	,		
29 1D 035 GS (group separator) 61 3D 075 = = 93 5D 135]] 125 7D 175 } } 30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ 31 1F 037 US (unit separator) 63 3F 077 ? ? 95 5F 137 _ _ 127 7F 177 DEL			" ' ' " ' '
30 1E 036 RS (record separator) 62 3E 076 > > 94 5E 136 ^ ^ 126 7E 176 ~ ~ 31 1F 037 US (unit separator) 63 3F 077 ? 2 95 5F 137 _ _ 127 7F 177 DEL	, ,,		
31 1F 037 US (unit separator) 63 3F 077 ? 2 95 5F 137 _ _ 127 7F 177 DEL			
	,,		
Source: www.LookunTables.com	31 1F 037 US (unit separator)	63 3F 077 ? ?	95 5F 137 _ _ 127 7F 177 DEL
Cource: www.cookap rubles.com			Source: www.LookupTables.com

Example: Simple image

Binary Interpretation:

00	White	01	Orange
10	Blue	11	Black



10	10	10	10	10	10	10	10	
10	10	01	01	10	01	01	10	
10	01	11	00	01	01	10	10	
10	01	00	00	01	01	10	10	
10	10	01	01	10	01	01	10	
10	10	10	10	10	10	10	10	
(b)								

10101010	10101010				
10100101	10010110				
10011100	01011010				
10010000	01011010				
10100101	10010110				
10101010	10101010				
(c)					

Figure 2. The (a) image representation, (b) two-bit cell representation, and (c) byte representation of a simple fish image. *Dive Into Systems*

Number bases and unsigned integers

Recall: Decimal numbers

$$5163 = 5 * 10^3 + 1 * 10^2 + 6 * 10^1 + 3 * 10^0$$

What is the general formula?

Notation

Ob (zero-b) denotes a binary number, e.g Ob 1001

Ox (zero-x) denotes a hexadecimal number, e.g. 0xE3

0 (zero) denotes an octal number, e.g. 0644

d₀ denotes the lowest order bit

d_{N-1} denote the highest order bit

Other popular notation: $74_{10} = 7 * 10^1 + 4 * 10^0$

Hexadecimal

Base 16: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Compact way to represent binary numbers Octal (base 8) is sometimes used

Dec	Bin	Hex									
0			4			8			12		
1			5			9			13		
2			6			10			14		
3			7			11			15		

Exercise: Hexadecimal

Convert this binary number to hexadecimal 0b0100110001101111101001100

Convert this hexadecimal number to binary

0x8045EF

Hexadecimal to decimal

Convert 0x3CD0 to decimal

Decimal to hexadecimal

Idea: Compute digits from highest order to lowest order

Example: divide by 16⁴, then 16³, then by 16², etc

```
Each step i from N-1 to 0 do

ith digit = floor(input/16i)

input = input % 16i
```

Exercise: Convert 9742₁₀ to hexadecimal

Digit Place	Input	Divisor	Digit (/)	Remainder (%)
d4		16 ⁴		
d3		16 ³		
d2		16 ²		
D1		16 ¹		
d0		16 ⁰		

Decimal to binary

Can use the same approach, but there is an easier way: repeated division

Idea: Repeatedly divide by 2 and check the parity

```
Each step i from 0 to N-1 do

ith digit = input % 2

input = floor(input / 2)
```

Exercise: Convert 422₁₀ to binary

Unsigned integers

Numbers ranging from 0 to a positive max value

Example: Represent 4 using a 4-bit unsigned integer

Example: Represent 34 using a 1 byte unsigned integer (e.g. a char)

Unsigned integer ranges

What is the largest number that can be stored in 4 bits?

What is the largest number that can be stored in 4 bytes?

Signed integers

Modern systems use a method called two's compliment

highest order bit encodes the sign (0 -> positive; 1 -> negative)

advantage: allows pos/neg numbers to be treated the same in hardware

Two's compliment

Suppose we have N bits to represent a signed integer. The formula is

$$-(d_{N-1} \times 2^{N-1}) + (d_{N-2} \times 2^{N-2}) + ... + (d_2 \times 2^2) + (d_1 \times 2^1) + (d_0 \times 2^0)$$

^ note the leading negative sign for just the first term!

Example: What is 1001 interpreted as a *signed* integer? As an *unsigned* integer?

Exercise: Signed integers

Convert the following *signed* integer to decimal: 0b 0110

Convert the following *signed* integer to decimal: 0b 1111

Negation

To negate a two's compliment signed integer flip the bits add one

Example: Compute -5 as a 4-bit signed integer

Question

If we use N bits, what is the range of unsigned integers we can represent?

If we use N bits, what is the range of signed integers we can represent?

Binary addition

Works like decimal addition: we carry over values when we reach our max digit.

Example: Add 2 + 8 as 4 bit unsigned binary numbers

Binary addition

Possible outcomes when considering two binary digits and 1 carry digit

Inputs			Outputs		
Digit A	Digit B	Carry in	Sum	Carry out	
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Binary subtraction

Idea: X - Y is the same as X + (-Y)

Therefore, negate the second operand and then add

Example: Compute 7-2 as unsigned 4-bit integers

Unsigned Overflow

When we try to store a value too large to fit into a data type, we get **overflow.**

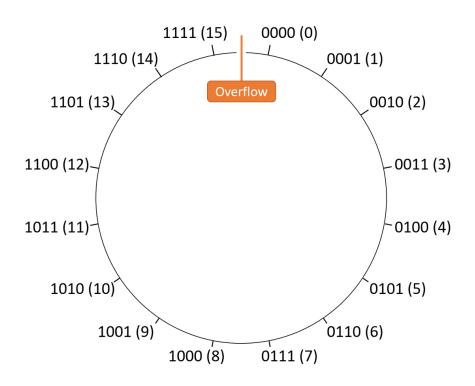


Figure 3. An arrangement of four-bit unsigned values into a modular space. All arithmetic is modular with respect to 2^4 (16).

Example: Add 12 + 7 as unsigned 4 bit numbers

Example: Add 2 - 3 as unsigned 4 bit numbers

Unsigned Overflow

Rule: If the carry-out doesn't match the carry-in, the computation has overflowed

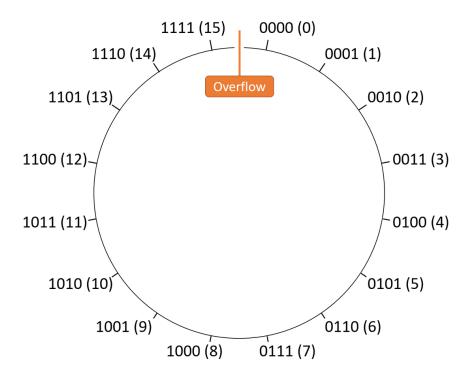


Figure 3. An arrangement of four-bit unsigned values into a modular space. All arithmetic is modular with respect to 2^4 (16).

When the carry-in = 0, we are adding and so the result should be *larger*. However, when the carry-out = 1, the result will be *smaller*.

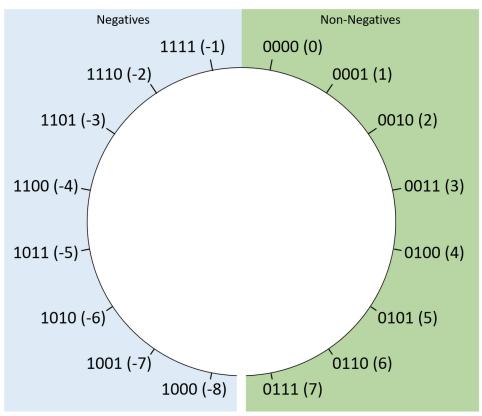
When the carry-in = 1, we are subtracting and so we want the result to be *smaller*. However, when the carry-out = 0, the result will be *larger*.

Demo: What is the output of this program?

```
#include <stdio.h>
int main() {
 unsigned int a = 0;
 for (a = 5; a >= 0; a--) {
   printf("Message!\n");
 return 0;
```

Signed overflow

When we try to store a value too large OR too small to fit into a data type, we get **overflow**.



Example: Compute -6 - 3 as signed 4 bit numbers

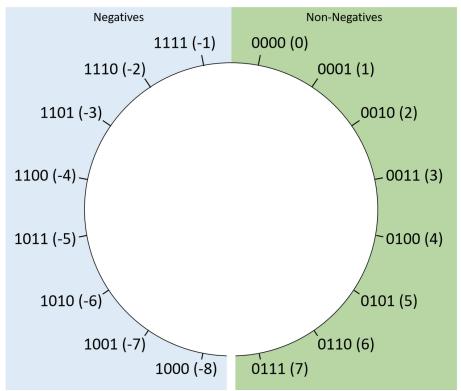
Example: Compute 5 + 3 as signed 4 bit numbers

Figure 36. A logical layout of two's complement values for bit sequences of length four.

Signed overflow

Rule: When the operands have different sign, overflow is impossible.

When the operands have the same sign, overflow occurs when the highest order bit of the result does not match the operand.



Idea: Moving towards zero is safe with signed integers.

Figure 36. A logical layout of two's complement values for bit sequences of length four.

Exercise: Compute 3 - 2 as signed 4-bit numbers

Exercise: Compute 3 - 4 as signed 4-bit numbers

Exercise: Add 7 + 3 as signed 4-bit numbers

Signed extension

What happens when you perform an arithmetic operation on numbers with different sizes? **signed extension**

For *unsigned* values, prepend 0

For signed values, prepend the leftmost bit

Example: Signed extension

Add 5 and -5 as 8 bit numbers