# ELE1601

Dr. Johnson

# Test 1 Practice

No calculator
No notes

Name _____

1) Write the code required to **declare** and **initialize** the following variables: 10 pts

A variable to hold the common letter grades(A, B, C, D, F). Initialize the variable to the lowest grade.

A variable to hold a 32bit signed value named foo. Initialize foo to 5.

Indicate the location of the first variable in Data Memory on the memory map

Data Memory

| | |
|---|---|
| xxxx xxxx | 1017 |
| xxxx xxxx | 1016 |
| xxxx xxxx | 1015 |
| | 1014 |
| | 1013 |
| | 1012 |
| | 1011 |
| | 1010 |
| | 1009 |
| | 1008 |
| | 1007 |


© tj

2 – Evaluate each of the following expressions – given: 10pts

```
int8_t a;    a = 5;
int8_t b;    b = 45;
int8_t c;    c = 14;
```

| | Decimal |
|---|---|
| b / c | |
| c % a | |
| -120 - 2 * a | |
| 3 * b | |
| b + + + - - a | |

3 – Add parenthesis to the following expression to indicate the proper precedence and associativity 10pts

$$z = x \mathrel{*}= a \mid\mid b \;\&\&\; c\texttt{++} \;\%\; 4 * d$$

4 – Evaluate each expression or statement and indicate the value and type of the result. 10pts

```
int a;      a = 4;     float c;    c = 1.6;     char e; e = 'a';
int b;      b = 12;    float d;    d = 0.5;     char f;  f  = 'C';
int foo;               float boo;               char soo;
```

foo = c / d

| Value | Type |
| --- | --- |
| | |

boo = b % (int) c + 3;

| Value | Type |
| --- | --- |
| | |

(int) d * c + d

| Value | Type |
| --- | --- |
| | |

5a) Provide the bit values for the bits indicated 6pts

S = 1001 1001

S[3] ☐

S[8] ☐

S[5:4] ☐

5b) Convert each number from KMG ←→ $2^n$ format (show your work) 4pts

16M → $2^{__}$

$2^{17}$ → _____


© tj

6 – Write each of the following numbers using the designated representation you must show your work . 10pts

60 → (unsigned 8bit binary)

binary

-90→ (8 bit 2's complement)

binary

2's complement 0110 1111 → (decimal)

decimal

1101 1101 → hex

hex

0xAD → 8 bit unsigned binary

binary

# 7 –Provide the arithmetic value and logical value for each expression

10 pts

```
int a;
int b;
float c;
char d;
a = 5;
b = 3;
c = 2.5;
d = 'f';
```

| | Arithmetic Value | Logical Value |
|---|---|---|
| b % a \|\| d | | |
| d \|\| b && 2 * c - a | | |
| (a \|\| b) + c | | |


© tj

# 8 – Provide the missing lines of code

10 pts

```
/*
 * test1_code.c
 *
 *  Created on: Mar 23, 2023
 *      Author: johnsontimoj
 */

int main(void){
    setbuf(stdout, NULL);

    // declare 2 integers a and b, a float c, and a char d

    // initialize the variable d to the letter Q

    // read in a value for the variable c
    printf("Please enter a float to store into variable c: ");

    // print the value of variable b

    return 0;
}// end main
```

9 – Provide a **code snippit** to do each of the following 10 pts

given: char a int b float c

a) Using 1 line of code: Print variables a and b on a single line together and c on a separate line

b) Using 3 lines of code: Read in variables c, b, and a (in that order)

10 – Provide a **code snippit** to do the following 10 pts

Don't write the whole program, just what is necessary

Declare, request and read in 3 variables, an int, a float and a character.
Calculate the average of the 3 variables and print the result

e.g. *The average of 12, 3.3, and G is 28.76666*

# C – Operator Precedence

| Precedence | Operator | Description | Associativity |
|---|---|---|---|
| 1 | ++ -- | Suffix/postfix increment and decrement | Left-to-right |
| | () | Function call | |
| | [] | Array subscripting | |
| | . | Structure and union member access | |
| | -> | Structure and union member access through pointer | |
| | (*type*){*list*} | Compound literal(C99) | |
| 2 | ++ -- | Prefix increment and decrement | Right-to-left |
| | + - | Unary plus and minus | |
| | ! ~ | Logical NOT and bitwise NOT | |
| | (*type*) | Type cast | |
| | * | Indirection (dereference) | |
| | & | Address-of | |
| | sizeof | Size-of | |
| | _Alignof | Alignment requirement(C11) | |
| 3 | * / % | Multiplication, division, and remainder | Left-to-right |
| 4 | + - | Addition and subtraction | |
| 5 | << >> | Bitwise left shift and right shift | |
| 6 | < <= | For relational operators < and ≤ respectively | |
| | > >= | For relational operators > and ≥ respectively | |
| 7 | == != | For relational = and ≠ respectively | |
| 8 | & | Bitwise AND | |
| 9 | ^ | Bitwise XOR (exclusive or) | |
| 10 | \| | Bitwise OR (inclusive or) | |
| 11 | && | Logical AND | |
| 12 | \|\| | Logical OR | |
| 13 | ?: | Ternary conditional | Right-to-Left |
| 14 | = | Simple assignment | |
| | += -= | Assignment by sum and difference | |
| | *= /= %= | Assignment by product, quotient, and remainder | |
| | <<= >>= | Assignment by bitwise left shift and right shift | |
| | &= ^= \|= | Assignment by bitwise AND, XOR, and OR | |
| 15 | , | Comma | Left-to-right |

# ASCII 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 | / |
| 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 | _ |
| 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] |