CS31 Worksheet: Week 2: C basics

**Discussion Block 1**

Q1. There is no boolean type in C, instead integer expressions used in conditional statements are interpreted as true or false according to this rule:

0: is false    non-zero value: is true

```c
int x, y;
x = 4;
y = -10
```

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Evaluates to: (T/F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>if (x &lt; y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>if (y)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>if (0)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For Loops

Q2. What does this for loop print?

```c
int arr[5];  // an array of 5 integers
float rates[40];  // an array of 40 floats
for (i=0; i < 5; i++) {
    arr[i] = i;
    rates[i] = arr[i]*2;
}
```

OUTPUT HERE:
Q3. Consider the following array layout in memory for an integer array “january_temps” that has 31 buckets.

```c
int january_temps[31];
```

What happens if we try to print `january_temps[35]`?

A) Error message because it is out of bounds of the array  
B) It’s 0 because it is out of bounds of the array  
C) It’s a garbage value because C doesn’t care, it’s your problem as a programmer to not ask for random offsets…  
D) Something else, list here: __________________________________________

Q4. Given what we know about arrays, how can we add a temperature reading to the second element in the array using the same library functions (read_int and read_float) as in Lab 1 from a text file?

A) `read_float (january_temps);`  
B) `read_float(&january_temps[1]);`  
C) `read_float(&january_temps[2]);`
Q5. Draw the stack diagram for the following code

```c
int func(int a, int y, int my_array[]) {
    y = 1;
    my_array[a] = 0;
    my_array[y] = 8;
    return y;
}

int main() {
    int x;
    int values[2];
    x = 0;
    values[0] = 5;
    values[1] = 10;
    x = func(x, x, values);
    printf("%d, %d, %d", x, values[0], values[1]);
}
```
Q1. Consider the following partial program:

```c
#include <stdio.h>

struct personT {
    char name[32];
    int age;
    float heart_rate;
};

int main(void) {
    struct personT p1;
    struct personT people[40];
    return 0;
}
```

(1) What type is each of the following expressions?

<table>
<thead>
<tr>
<th>expression</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td></td>
</tr>
<tr>
<td>p1.name</td>
<td></td>
</tr>
<tr>
<td>p1.heart_rate</td>
<td></td>
</tr>
<tr>
<td>people</td>
<td></td>
</tr>
<tr>
<td>people[0]</td>
<td></td>
</tr>
<tr>
<td>people[0].name</td>
<td></td>
</tr>
<tr>
<td>people[0].name[3]</td>
<td></td>
</tr>
</tbody>
</table>

(2) Write the C code to set the age of the 3rd person in the people array to 18:
Q2. Data representation:

A number, written as the sequence of N digits $d_{n-1}d_{n-2}...d_1d_0$ where $d$ is in $\{0,1\}$, represents the value:

$$[d_{n-1} \times 2^{n-1}] + [d_{n-2} \times 2^{n-2}] + ... + [d_2 \times 2^2] + [d_1 \times 2^1] + [d_0 \times 2^0]$$

i) What is the value of $0b110101$ in decimal?

Options: a) 26  b) 53  c) 61  d) 106  e) 128

ii) What is the value of $0x1B7$ in decimal? (Note: $16^2 = 256$)

$$[d_{n-1} \times 16^{n-1}] + [d_{n-2} \times 16^{n-2}] + ... + [d_2 \times 16^2] + [d_1 \times 16^1] + [d_0 \times 16^0]$$

<table>
<thead>
<tr>
<th>DEC</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>HEX</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td>E</td>
<td>F</td>
</tr>
</tbody>
</table>

Options: a) 397  b) 409  c) 419  d) 437  e) 439

iii) Converting Binary to Hexadecimal

Each hexadecimal has 16 possible values (= $2^4$ bits of information).
Therefore 1 hexadecimal = 4 bit value

$0x1B7$ in binary = 0001 1011 0111

    1   B   7

Now, try converting $0b0011 1100 1010 1101 1011 0011$ to Hexadecimal