CS31 Worksheet: Week 2: C basics

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

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

Expression	Value	Evaluates to: (T/F)
if (x < y)		
if (y)		
if (0)		

An Example with Local Variables

```
/* a multiline comment:
   anything between slashdot and dotslash */

#include <stdio.h> // C's standard I/O library (for printf)

int main() {
   // first: declare main's local variables
   int x, y;
   float z;

   // followed by: main function statements
   x = 6;
   y = (x + 3)/2;
   z = x;
   z = (z + 3)/2;

   printf(…)  // What is the output here?
}
```

Clicker choices

	X	Y	Z
A	4	4	4
B	6	4	4
C	6	4.5	4
D	6	4	4.5
E	6	4.5	4.5

While Loops

Basically identical to Python while loops:

```
while(<boolean expr>) {
    while-expr-true-body
}
```

Which one of these results in an infinite loop?

A) while (x <100)
B) while(1)
C) Both A and B
D) Neither A or B

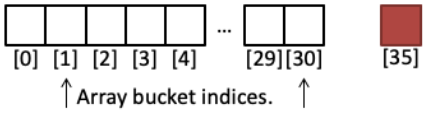
```
x = 20;
while (x < 100) {
  y = y + x;
  x += 4;  //  x = x + 4;
}
<next stmt after loop>;
```

```
x = 20;
while(1) {
  y = y + x;
  x += 4;
  if(x >= 100) {
     break;  // break out of loop
  }
}
<next stmt after loop>;
```

Given what we know about arrays, how can we add a temperature reading second element in the array?

```
int january_temps[31];  // Daily high temps
```

"january_temps"
Location of [0] in memory.



1. scanf("%d", january_temps);
2. scanf("%d", &january_temps[1]);
3. None of the above

For Loops

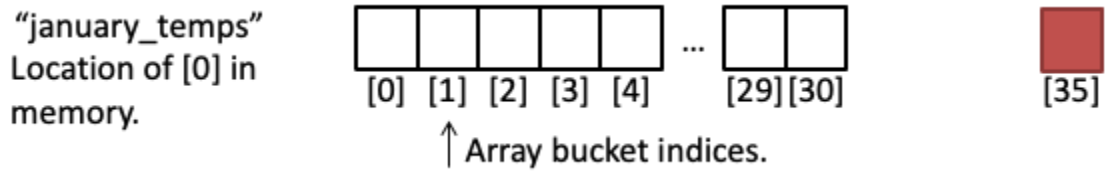
Q2. What does this for loop print?

```
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.

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]);