

Answer the following questions and show all work. I can't give partial credit if you get an answer wrong and don't show any work.

1. C has a data type called a **short** that is a two byte integer on the Raspberry Pi. For example, you can declare a variable to be either a **signed** or **unsigned short**. Give the declarations below ...

```
short x;  
unsigned short y;
```

- a. [3] What is the smallest possible value of **x**, expressed in decimal.
- b. [3] What is the smallest possible value of **x**, expressed in binary.
- c. [3] What is the smallest possible value of **x**, expressed in hexadecimal.
- d. [3] What is the largest possible value of **y** expressed in binary.
- e. [3] The **%x** modifier in a **printf** format string will print an integer in hexadecimal. What will the following C statement print? `printf("%x", -1);`
- f. [3] What would be printed by `printf("%d", sizeof(x));`
- g. [5] What is printed by the following program?

```
#include <stdio.h>  
int main() {  
    signed char x, y, z;  
    x = 90;  
    y = 88;  
    z = x + y;  
    if (z > x)  
        printf("Foo\n");  
    else  
        printf("Bar\n");  
}
```

Output: \_\_\_\_\_

2. [5] What is the output of the following program? Watchout! This is very similar to a study question. But not identical.

```
#include <stdio.h>
int main() {
    int s = 0;
    int n = 40;
    while (n > 0) {
        s = s + !(n & 1);
        n = n >> 1;
    }
    printf("%d\n", s);
}
```

Output: \_\_\_\_\_

3. [2] The **&** operator applied to a variable (as in **&x**) is called the \_\_\_\_\_ operator.
4. [2] Adding two positive integers with the result being negative is called \_\_\_\_\_.
5. [2] If  $2^x = 1024$  then **x** must be \_\_\_\_\_.
6. [2] Express **-33** as an 8 bit two's complement integer. Show all work.
7. [2] **0xDeafBeef** is a valid C hexadecimal constant. True/False    Answer: \_\_\_\_\_
8. [5] Write a very short C code fragment that declares a variable **x** to be an integer and **p** to be a pointer to an integer. Have **p** point to the integer **x**.
9. The formula for converting fahrenheit to celsius is  $c = (f-32)5/9$ .
- [5] Make a directory **exam1** in your course repository.
  - [5] In the **exam1** directory, create a header file **f2c.h** that declares a function named **f2c** that takes a double and returns a double.
  - [10] In the **exam1** directory create a file **f2c.c** that implements the function **f2c**.
  - [10] In the **exam1** directory create a file **main.c** that takes a command line argument (the `argc`, `argv` stuff) and prints the argument converted to celsius. The function **atof** declared in **stdlib.h** converts a string to a double.
  - [5] Push the files **f2c.h**, **f2c.c**, and **main.c** to your GitHub repository. Log in to GitHub to make sure they are there. Do not modify the files after they are pushed. They are timestamped.