

HW 2 – Fixing function `int2bin`

Due Date: Monday, January 7 by 9:30AM.

In this homework you are going to fix the version of `int2bin` that printed the binary representation of a string in reverse (the wrong order). For example `int2bin(43)` printed 110101 (incorrect) instead of 101011 (correct).

```
// incorrect version of int2bin we wrote in class
void int2bin(int n) {
    while (n > 0) {
        printf("%d", n % 2);
        n = n / 2;
    }
    printf("\n");
}
```

One way to fix this is to use a string in C. A string is just an array of characters where the last character is the special control character `'\0'`. This control character terminates the string. For example:

```
char str[10];    // str is an array of ten characters
str[0] = '1';
str[1] = '0';
str[2] = '1';
str[3] = '\0';  // terminate the string with '\0'
```

You can print strings with the `%s` format specifier.

```
printf("str is %s\n", str); // prints 101 and a newline
```

You print a single character using the `%c` format specifier.

```
printf("%c", str[0]); // print the first character in str
```

Part 1 [10 points] Fix the `int2bin` function so that the functions prints the binary representation of an integer in the correct order.

Part 2 [10 points] Write a main program that prompts the user to enter an integer to convert to binary, and continues until they enter a negative value. Here is some sample output of my version. Your output should look similar.

```
ehar@raspberrypi:~/ehar_CS220/hw2 $ ./i2b
```

```
Enter an int: 23
23 = 10111
```

```
Enter an int: 3
3 = 11
```

Enter an int: 5
5 = 101

Enter an int: 9
9 = 1001

Enter an int: 200
200 = 11001000

Enter an int: 123456789
123456789 = 111010110111100110100010101

Enter an int: -1
Bye.

What to turn in

Create a directory named **hw2** in your repository that contains file **main.c**, **int2bin.c**, and **int2bin.h**.