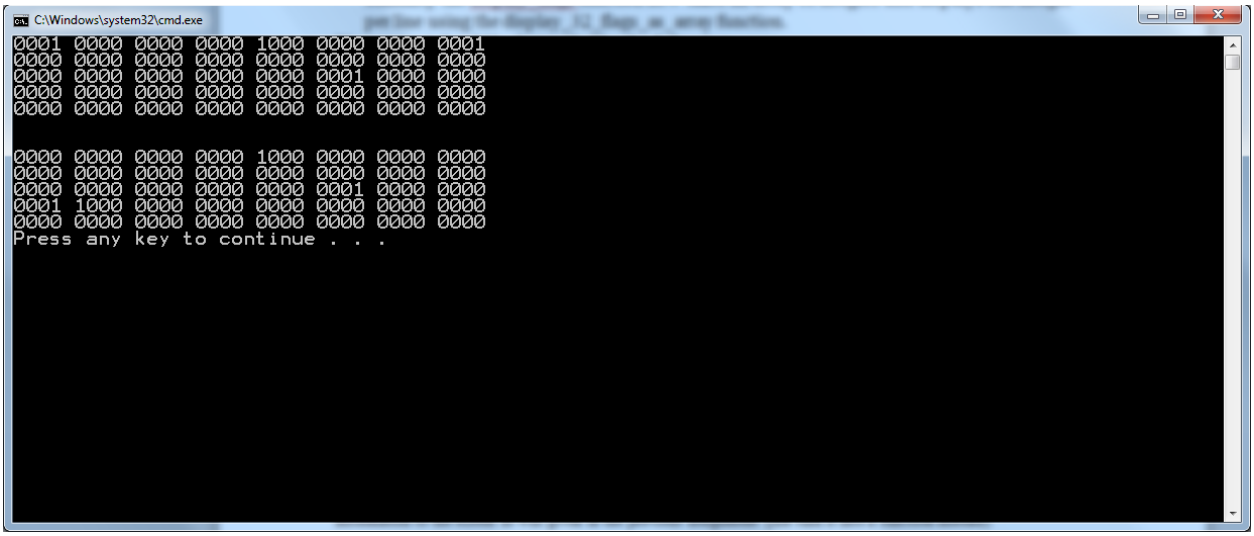
This daily will allow you to practice more with the bit wise operators and shifts. Consider the following modification of the main program from daily 4:

```
void set_flag(unsigned int flag_holder[], int flag_position);
void unset_flag(unsigned int flag_holder[], int flag_position);
int check_flag(unsigned int  flag_holder[], int flag_position);
void display_32_flags_as_array(unsigned int flag_holder);
void display_flags(unsigned int flag_holder[], int size);

int main(int argc, char* argv[])
{
	unsigned int flag_holder[5] = { 0 };//Set the first integer to zero and all others
to zero by default.

	set_flag(flag_holder, 3);
	set_flag(flag_holder, 16);
	set_flag(flag_holder, 31);
	set_flag(flag_holder, 87);

	display_flags(flag_holder, 5);
	printf("\n\n");

	unset_flag(flag_holder, 31);
	unset_flag(flag_holder, 3);
	set_flag(flag_holder, 99);
	set_flag(flag_holder, 100);

	display_flags(flag_holder, 5);
	return 0;
}
```

Here I have changed the functions so that they take an array of integers instead of just one integer. This allows me to imagine that I have a long array of bits instead of an array of integers. The functions can now set, unset, check and display flags for any bit in the array of 5 integers that I have made (and should work for any size array as long as your bit index is in bounds of your array).

I also changed the display behavior. Daily 4 displayed the flags as you would see them in a binary number but since this program is moving away from the idea of a binary number to store bits and moving toward the idea of having an array of bits the display_32_flags_as_array function will display the [0] bit first then [1] and so on up to 31 whereas the display_32_flags function in daily 4 displays the [31] bit first and down to [0]. Similarly the display_flags function now takes an array of integers and displays one integer per line using the display_32_flags_as_array function.

Your output should look exactly like the following:



You may want to be careful about how you call your check_flag function from inside the display_32_flags_as_array function since that function receives an integer and check_flag is expecting an array. How can you overcome this obstacle?

At the top of your code you should have a comment section that has the following format:

```
/*************************************************************
       Author: <your name>
       Date: <Today's date>
       Effort: <Time you spent on this project>
       Purpose: <Purpose of this assignment in your own words>
*************************************************************/
```