The University of New Mexico

CS 241—Data Organization

Lab 3: Bitwise Crossover

Description:

Write a function:

```
    unsigned short crossover(
      unsigned short a, unsigned short b, int n)
```

that performs a very small part of a genetic algorithm by returning the integer composed of the lower **n** bits of **a** and the upper **16-n** bits of **b**.

Note: this assumes that **sizeof(unsigned short)** is 2 bytes. This is the case on moons.unm.edu.

For example:
crossover(43690, 18724, 6) would return **x=18730** since:

	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
a	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
b	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0
x	0	1	0	0	1	0	0	1	0	0	1	0	1	0	1	0

Write a function, **main**, that reads characters from the standard input stream (stdin). Each record of input should consists of three, space delimited numbers and be terminated by **'/n'**.

Your main function must continue reading from stdin until EOF is encountered.

Your program must error check for:

1. Numbers out of range,
2. Illegal characters (any non-digit character and non-space character)
3. Any record that does not contain exactly 3 space delimited numbers.

Your program must produce one line of output for each line of input. Each line of output must either be the correct return value of crossover or **"error".**

Hint: For reading in the numbers, use **atoi** (ASCII to Int). This is a standard C library function. Its source code is developed in the textbook in section 2.7. The **atoi** function expects the input ASCII digits to be in a character array. On the class website, you can find **readNumbers.c**, an example program showing how to read each line of input and convert sequences of ASCII digits on that line to integers and save them in an array of integers. I have also posted a sample input file: **readNumbers.in**.

Hint: For debugging of this lab, it will often be helpful to see the hexadecimal values of fields. The following code will print anything as an unsigned hexadecimal integer:

```
printf("%X\n", a);
```

Turning in your assignment

Attach your program file **crossover_*yourname*.c** in BlackboardLearn.

Grading Rubric (total of 20 points)

[-2 point]: The program does not start with a comment stating the students first and last name and/or the source file is not named correctly.

[-2 points]: Program compiles with warnings on **cs.unm.edu** using **/usr/bin/gcc** with no options.

[-5 points]: Code does not follow the CS-241 standard.

[12 points]: Passes diff test with twelve known tests cases given in crossover.in (on class website) with **crossover.out**. One point each test. NOTE: DO NOT copy/paste these files from a browser. DOWNLOAD and save, or use the Linux cp command to copy to your directory on the cs machines. When you copy/paste a file, you will likely get the wrong line ending characters.

[8 points]: Passes 8 unknown data tests. These will not be wildly different from the known tests, mostly just different numbers, but some may trigger a bug not found in the known 12.