**Exercises-NumberRepresentation**

**Description**

In this exercise, we will practice converting between radices and using two’s-complement arithmetic. Before beginning, review the class information on binary, octal, decimal, and hexadecimal numbers and on how two’s-complement works.

To check your results, you can use a combination of the Unix command-line (bash) function `printf` and a program I provide named `binout`.

The part of this exercise to be transferred is towards the end. They mimic some problems from the text.

`printf` can be used to convert between decimal, octal, and hexadecimal numbers. You can also output a negative decimal number in hexadecimal and octal. Because all octal input numbers are assumed to be unsigned, you cannot give `printf` a two’s complement number in hexadecimal or octal and expect it to output it as a negative decimal number.

`printf "outputformat" inputnumber` output the `inputnumber` using `outputformat`. `outputformat` must be enclosed in double-quotes and it should always end in `\n` (to output a newline). The formats are:

- `%x` output in hexadecimal
- `%d` output in decimal
- `%o` output in octal

The `inputnumber` is, by default, decimal. Decimal numbers may be signed. Signed decimal numbers output in octal or hexadecimal are output as 64-bit numbers. The `inputnumber` is interpreted as octal if the first digit is `0`. It is interpreted as hexadecimal if the first characters are `0x`.

**Examples**

- Output the number 47 in hexadecimal
  
  ```
  -bash$ printf "%x\n" 47
  2f
  ```

- Output the number −47 in hexadecimal
  
  ```
  -bash$ printf "%x\n" -47
  ffffffff0fffffff1
  ```

  In the output above, you simply ignore the leading digits that you don't want. If you want a 32-bit hexadecimal value, ignore the first 8 f’s, resulting in `0xfffffffd1`.

- Output the octal number 7456 in decimal
  
  ```
  -bash$ printf "%d\n" 07456
  3886
  ```

`binout` can be used to convert signed decimal and unsigned octal and hexadecimal to 32-bit binary values. `binout` can be found as `bin/binout` relative to the public work area.

**Examples**

- `bash$ pwd` /pub/cs/gboyd/cs270/bin
- `bash$ ./binout 0xff`
  
  ```
  hex = 0x000000FF
  bin = 00000000000000000000000011111111
  ```
- `bash$ ./binout 1234`

This document was produced with free software: OpenOffice.org on linux.
hex = 0x000004D2
bin = 00000000000000000000010011010010

bash$ ./binout -4
hex = 0xFFFFFFFC
bin = 11111111111111111111111111111100

bash$

The Exercise

On a sheet of paper, fill in the following table with the missing values. Check your work using printf and binout as indicated above. Do some values of your own until you feel proficient.

<table>
<thead>
<tr>
<th>binary</th>
<th>hexadecimal</th>
<th>octal</th>
<th>decimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>111111111111111111111111101110</td>
<td>0xf0</td>
<td>01175</td>
<td>-89</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1025</td>
</tr>
<tr>
<td></td>
<td>0x1356</td>
<td></td>
<td>-4383</td>
</tr>
</tbody>
</table>

Problems to transfer

Do problems 3.1, 3.4, 3.6, 3.7, and 3.8 from your text. It is not clear, but you must begin each by encoding the starting number in binary, then interpreting it as indicated. Thus, for problems 3.6-3.8, start by encoding 185 and 122 in 8-bit [unsigned] binary. Then interpret the bit patterns as indicated for each question (unsigned for 3.6, 2's complement for 3.7 and sign and magnitude for 3.8).

Create a text file with your answers to these questions (with the problem numbers) named nr
Transfer nr to get credit for this exercise set. Justify your answers. (Note that the name must be exactly two characters long: nr - not nr.txt or ...)

A thought question

These are very good to pose on the Google Group.

George has a garage-door opener with a keypad that takes a four-digit number. He gave the code to Sue so that she could walk the dog when he is not home, but when Sue opens the keypad, she cannot remember the number. She notices, however, that only the 3 and 6 digits on the keypad have smudges indicating those are the only numbers that have been used repeatedly.

a. Assuming these two digits are the only digits in the four-digit code, what is the maximum number of combinations Sue must try before she can open the garage door?

The answer should be a warning to all of you who use keypads: clean them often!

Hint: the answer is not the obvious one (16). Why?

b. Suppose the code has four different digits, but the keypad smudges indicate what the digits are. What is the maximum number of combinations now? If it takes 5 seconds per attempt, what is the average amount of time it would take to open the garage door? Do you think a neighbor would have noticed by then?

Answers and Hints

You can check your number conversions using the above tools. Do so. Check the problems from the book and the Thought Questions on the Google Group with classmates. You might want to ask about 'sign and magnitude' format.