University of Arkansas
Department of Computer Science and Computer Engineering
2006 High School Programming Contest
Problems

Revision Date: February 19, 2007

Note that for all problems, an integer is defined as:

an optional minus sign,

followed by one of the digits from the set 0..9,

followed by any number of digits from the set 0..9

Thus `3`, `-21`, and `309` are integers; `- 4` and `+38` are not.

1 Ulam's Problem: finding cycle length

The $3n + 1$ conjecture, due to Collatz and also known as Ulam's problem, states that any positive integer n ($n > 2$), when repeatedly operated upon the following rule, eventually yields 1.

If the n is odd, let n be $3n + 1$, else let n be $\frac{n}{2}$.

For example, if n starts out as 27, then the first few values of n are:

```
27 82 41 124 62 31 ...
```

The number 82 is obtained since 27 is odd. Thus, one multiplies 27 by three and adds 1, yielding 82. The number 41 is obtained since 82 is even. Thus one divides 82 by 2, yielding 41.

Your task is to read in a single positive integer from a file and give the number of iterations of the rule required to reach 1. For example, if the number read from the file is 2, it should be reported that the number of iterations required is 1.

input file input; the name of the file is passed as a command line argument; the file contains a single positive integer surrounded by white space;

output the required number of iterations is reported to the console (stdout)

2 Ulam's Problem: Finding n

Given a cycle length c and a set of bounds representing a range r, display the smallest value of n within r whose $3n + 1$ cycle length is c. If no number in the range has the given cycle length, display 0.

input file input; the name of the file is passed as a command line argument; the file contains three positive integers separated by white space; the first integer is the cycle length c; the second integer is the low end of the range r; the third number is the high end of the range r; the bounds are inclusive (*i.e.*, both the low and the high ends should be checked)

output the smallest integer in the range is reported to the console (stdout)

3 Encryption: simple decryption

Before the internet, there was *Usenet*. Usenet was kind of an internet chat group that communicated via email. One of the most popular newsgroups (as the groups were called) was rec.humor.funny. Due to the fact that it was easy for a reader inadvertently to glance at the punchline before finishing the setup, the punchline was encrypted. Most news readers at the time incorporated a command to decrypt any encrypted text.

Usenet users encrypted text with a version of the Caeser cipher, which is a one-to-one substitution cipher. For Usenet, the ROT13 cipher was used: each letter was rotated 13 letters to the right. Thus the letter 'a' was encrypted as the letter 'n' while the letter 'b' was encrypted as the letter 'o', the letter 'm' was encrypted as the letter 'z' and so on. The rotation was circular, in that the letter 'n' was encrypted as the letter 'a' and 'z' was encrypted as the letter 'm'. The fact that 'a' is encrypted as 'n' and vice versa is due to the fact that there are 26 letters in the English alphabet and that the rotation of 13 is exactly half that. With a different number of letters in the alphabet or a different rotation number, this symmetry would not be seen.

Uppercase letters are treated the same way. The letter 'A' is encrypted as the letter 'N' and the letter 'Z' is encrypted as the letter 'M'. All other characters are left unchanged.

Given a ROT13 message, display the decrypted text.

input file input; the name of the file is passed as a command line argument; the file contains an arbitrary amount of text encrypted with a ROT13 cipher

output the decrypted text is reported to the console (stdout)

4 Encryption: decryption using a crib

During WWII, the scientists at Bletchley Park, England were successful in breaking the Nazi Enigma cipher due, in part, to the fact that encrypted messages from submarine commanders always started with information about the current state of the weather. Using actual weather reports, the scientists (including Alan Turing, considered the father of Computer Science) were able to reconstruct some of the unencrypted text (or plaintext). Knowing a portion of the plaintext is a powerful tool for breaking a substitution cipher.

Your task is to break a ROTN cipher. You are given the encrypted text, which has been encrypted by rotating each letter N letters to the right and a portion of the plaintext, known as a *crib*. Your task is to *find the rotation* N *that was used to encrypt the plaintext.*

You may find more than one value of N since more than one decryption may yield a plaintext that contains the given crib.

Given the cyphertext and a crib, print all values of N that could have been used to encrypt the plaintext. For each value of N, print the associated decrypted text. Of course, each decrypted version of the encrypted text should contain the crib.

input file input; the name of the file is passed as a command line argument; the file contains two lines; the first line is the encrypted text; the second line contains the crib; there are no leading or trailing spaces in either the encrypted text or the crib

output for each possible N whose associated decrypted text contains the crib, report N followed by the decrypted text to the console; N may range from 0 to 25, inclusive.

5 Playing with numbers: reversing digits

Reverse a decimal (*i.e.*, base 10) integer. For example, the number 1234 should produce the number 4321. The number 120 should produce the number 21.

input file input; the name of the file is passed as a command line argument; the file contains a single integer on the first line; an integer is a contiguous span of the digits '0' through '9'; there are no leading or trailing spaces

output report the original integer and its reversal to the console

6 Playing with numbers: producing palindromes

There is a peculiar property about reversing a decimal integer. If we iterate the following action...

Let n be $n + reverse(n)$

...eventually n will be a palindrome (i.e., reading the digits from left to right is the same as reading the digits from right to left). For example, 1221 is a palindrome as are 3, -33, 12321, and 123321. The numbers 34, 556, and -90 are not palindromes.

Given an original integer n and a limit on the number of iterations t, report the original integer, the resulting palindrome, and the number of iterations required if the palindrome can be found within t number of iterations. If the original integer is already a palindrome, report the original integer and a message stating that it already is a palindrome. Otherwise, report the original integer followed by a message stating that a palindrome could not be found.

input file input; the name of the file is passed as a command line argument; the file contains two integers separated by white space; the first integer is the original value of n; the second integer is the limit on the number of iterations; the limit is positive and inclusive (for example, if the limit is 5, at most five iterations would be attempted).

output report, to the console, the original integer and *either* the resulting palindrome and the number of iterations required, *or* the fact that it already is a palindrome, *or* the fact that a palindrome could not be found within the limit.

7 Number bases: converting base 36 to decimal

The symbol set...

```
0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

...can be used to represent base 36 numbers, where A is the digit corresponding to 10 and Z is the digit corresponding to 35. The decimal (base 10) equivalent of 1CY (base 36) is

$1 * 36^2 + 12 * 36^1 + 34 * 36^0$

or 1762. The 12 and the 34 come from the fact that C is the 12^{th} digit and Y is the 34^{th} digit.

Given a number in base 36, determine its decimal equivalent and report it using the form:

```
1CY (base 36) is 1762 (base 10)
```

input file input; the name of the file is passed as a command line argument; the file contains a single number in base 36

output report the original base 36 number followed by its decimal (base 10) equivalent to the console

8 Number bases: adding numbers in arbitrary bases

Given a number set of n unique, printable characters on the first line representing the successive digits of a base n numbering system and subsequently two numbers in that base, add the two numbers together and report the sum in that base. For example, if the symbol set is...

```
0123456789
```

...and the subsequent numbers are...

```
27 31
```

...you would report:

```
base 10: 27 plus 31 equals 58
```

If the symbol set is...

```
01234567
```

...and the subsequent numbers are...

```
27 31
```

you would report:

```
base 8: 27 plus 31 equals 60
```

As a final example, if the symbol set is...

```
ABC,01234567EFG!
```

...and the numbers to be added are...

```
27 31
```

...you would report:

```
base 16: 27 plus 31 equals GA
```

In this last case, the G is the digit representing 14 and A is the digit representing 0.

input file input; the name of the file is passed as a command line argument; the file contains two lines; the first line contains the symbol set with no leading or trailing spaces; the symbol set is contiguous and symbols are drawn from the set of printable characters the second line contains two numbers in the base specified by the the symbol set; the two numbers are separated by whitespace.

output report the magnitude of the base, the two input numbers, and the sum, in that order, to the console