— Computist Quiz —

Just Four Chars. Let **`x`** be an integer, as in C++ or Java. Describe what the expression

```
x & - x
```

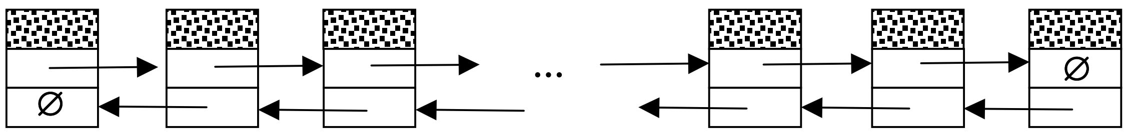
will compute.

Parody Bit. Let **`x`** be an integer larger than the odd number **`q`**. Change the value of **`x`** using the following rule

```
if  x  is even
  then  x / 2
  else  x – q
```

until **`x`** becomes smaller than **`q`**. If the final value of **`x`** is zero, what can you say about the original input value?

Duncing Lunks. A list that is equally likely to be traversed starting from either end is usually doubly-linked:



However in a certain application footprint is at a premium. Describe an implementation that uses only half as much space to store link information:

yet can be traversed in either forwards or backwards with equal efficiency, no matter which direction is chosen.

++Unshuffle. To *unshuffle* a collection of 2^n objects you move all the odd-index elements so they follow all the even-index elements, while otherwise preserving their order. For example, with $n = 3$:

`A0 A1 A2 A3 A4 A5 A6 A7` → `A0 A2 A4 A6 A1 A3 A5 A7`

Suppose you start by unshuffling a collection of 1024 elements. Then you unshuffle the first 512 elements, and follow it by an unshuffle of the second 512 elements. Next, you unshuffle each of the four chunks of size 128, and continue, successively halving the chunk size (and doubling the number of chunks). When you get done…

How many elements wind up back in their original positions?
Which elements are they?
Which elements wind up the furthest from their original positions, and how far away are they?
What would happen to the collection if you were to apply a *second* pass of the entire procedure?
Can you suggest any possible applications for this procedure?

Siamese Sequences. Describe a relationship between the following sequences:

```
0 1 3 3 7 7 7 7 15 15 15 15 15 15 15 15 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 31 63…
0 1 3 0 4 1 7 0  8  1 11  0 12  1 15  0 16  1 19  0 20  1 23  0 24  1 27  0 28  1 31  0 32…
```


fixpoint fxpt.com Inc.

41119

For answers eMail :
info@fxpt.com

— Computist Quiz —

Sordid Sort. Often a system string comparison sorts alphanumeric data "unnaturally", as with these filenames:

```
File1
File10     ← ?
File11     ← ?
File12     ← ?
File2
File3
File3X
File4
File5
File6
File7
File8
File999
```

Describe an algorithm that will sort alphanumeric strings in a "more natural" order:

```
File1
File2
File3
File3X
File4
File5
File6
File7
File8
File10
File11
File12
File999
```

The algorithm may only rely on comparing characters; to avoid overflow problems with very long numbers it must *not* convert substrings of digits into their numerical values.

Scary Natter. A mask programming error in a chip accidentally creates *a binary adder whose carries skip left an extra bit*, producing a computer with an "add" instruction that computes the following peculiar outputs:

⊕	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	…
0	0	1	2	3	4	5	6	7	8	
1	1	4	3	6	5	16	7	18	9	
2	2	3	8	9	6	7	12	13	10	
3	3	6	9	12	7	18	13	24	11	
4	4	5	6	7	16	17	18	19	12	
5	5	16	7	18	17	20	19	22	13	
6	6	7	12	13	18	19	24	25	14	
7	7	18	13	24	19	22	25	28	15	
8	8	9	10	11	12	13	14	15	32	
…										…

Describe a model for *what* this ⊕ operation computes (versus the above description of *how* it computes it). Can you suggest any possible applications for this operation?

fixpoint fxpt.com Inc.

41119

For answers eMail :
`info@fxpt.com`

— Computist Quiz —

Review. Let **0 ≤ x ≤ 7**. Which other question in this quiz is most related to the integer expression below?

```
(((x * 65) & 322) % 15)
```

Amphibious Discursion. A predicate on positive integers:

```
boolean isToad(int n) {
  return n==2 || n==frog(4,floor((n-1)/2),1,0)-1);
}
```

is defined with the aid of the following helper method:

```
int frog(int q, int r, int s, int t) {
  if (r<t) {
    return 0;
  } else if (r==t) {
    return 1;
  } else if (q==0) {
    return 0;
  } else {
    return frog(q,r,s+1,s+t) + frog(q-1,r-t,1,0);
  }
}
```

Which integers are toads? Describe what the **frog** method does. Can you explain why this code works?

Distinctly Odd. Justify (or prove) the following theorem (Euler c. 1740)

The number of partitions of *n* into *distinct* parts

equals

the number of partitions of *n* into *odd* parts.

Can you give a purely visual demonstration, without using any words at all?

The Dismal (Computer) Science. Tired of brainiac quizzes, we adopt *dismal* arithmetic, which is like decimal, but has no carries. And, as we work through each pair of digits in a calculation, we just follow the lazy rules:
*to **A**dd, take the l**A**rger; to **M**ultiply, take the s**M**aller*:

```
  169
+ 248
  269
```

and

```
    169
  x 248
    168
   144
+ 122
  12468
```

For all integer **x**, what is the unique "dismal unit" **u** such that the dismal product **ux** always equals **x**?
A dismal prime **p** isn't the dismal product of any numbers except **u** and **p**. What are the dismal primes <100?

fixpoint fxpt.com Inc.

For answers eMail : **info@fxpt.com**

— Computist Quiz —

Two's Compliment. Given that negative one is represented in two's complement by an infinite string of **1**s:

…1111111111 = -1

describe what value is represented by an infinite string of *alternating* **0**s and **1**s:

…0101010101 = ?

Poppyseed Place. Which pair doesn't belong with the others?

+ *and* **-**
& *and* |
min *and* **max**
gcd *and* **lcm**

Guilt by Dissociation. Give an example of an operation which always commutes but isn't associative:

ab = ba *for all* **a,b**

but

a(bc) ≠ (ab)c *for some* **a,b,c**

Plugh & Plover. Essay the difference between *clever* and *smart*.

Nullicorn. Explain how one can both believe that unicorns have *one* horn, yet also that there are *zero* unicorns.

Terasecond. If you were magically transported back in time one trillion seconds, describe how you might go about creating a programmable computer.

Nanograph. Write an inspiring bedtime nanostory for computists. Nanofiction must have characters, setting, conflict, denouement—and be no more than 55 words long (including title).

fixpoint
fxpt.com Inc.
41119

For answers eMail :
info@fxpt.com