Introduction to Programming and Data Structure

Subject Code: CS1L001

Dr Sudipta Saha

Assistant Professor

Computer Science and Engineering,
School of Electrical Sciences
Indian Institute of Technology Bhubaneswar

What is a byte

What is a bit

Number system

Unary

Binary

Decimal

Hex

How to determine size of a variable using program ?

Number System

Why we need to Study ?

Why variable’s size varies ? Int , float, double …

How to quantify the size of a variable ?

What is the unit of the size of a variable ?

Number System

Need to understand how variables are stored …

Better to Say – are represented …

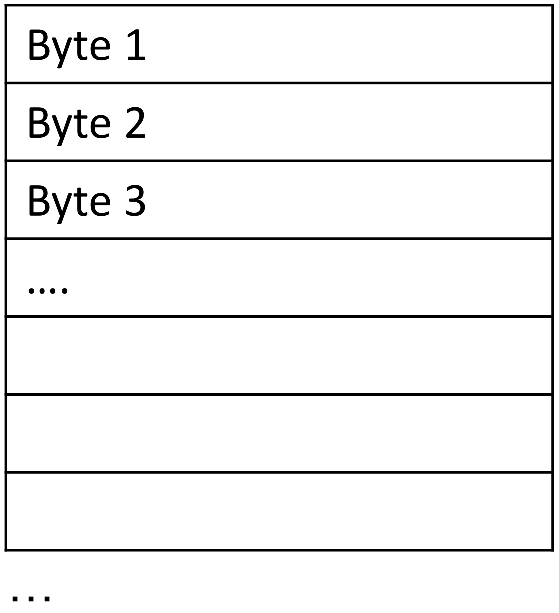
Number System

Need to understand how variables are stored …

Better to Say – are represented …

Variables resides in the memory

Memory is a collection of series of bytes



Number System

Need to understand how variables are stored …

Each Byte is composed of 8 bits

One bit is the abbreviation of **Binary Digit**

One byte

Bit 0	**Bit 1**	**Bit 2**	**Bit 3**	**Bit 4**	**Bit 5**	**Bit 6**	**Bit 7**

Number System

Need to understand how variables are stored …

So, the memory can be seen as -

…

Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7
Bit 0	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7

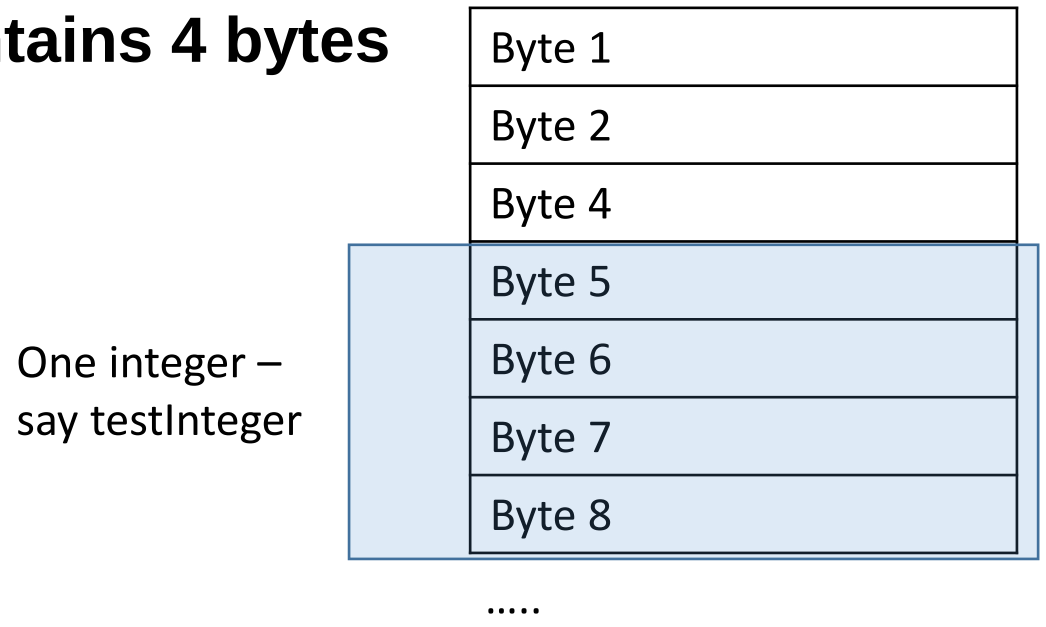
…

Number System

Need to understand how variables are stored …

A variable is composed of some bytes –

One integer usually contains 4 bytes



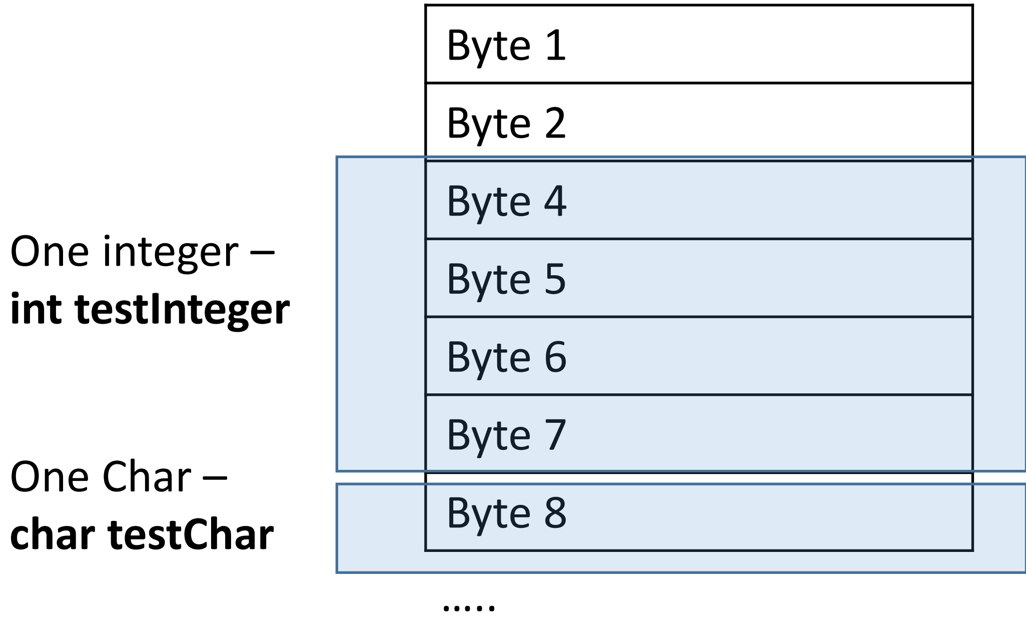
Number System

Need to understand how variables are stored …

A variable is composed of some bytes –

One integer usually contains 4 bytes

One char contains one byte only



Number System

What is a bit ?

Binary Digit

Binary means 2

Digit means kind of symbols

Binary digits are the symbols used in a binary ***number system***

* & ~ @

* * * *

Binary number systems –

Two symbols – 0 and 1

Number System

Why we need a number system ?

To *count, enumerate, quantify using number …*

Common use – how to count how many days have passed since I have seen you ? –

How was done in ancient days ?

Number System

The first number system –

Unary number system

1	|
11	| | ….
111	
11111111….	

Number System

The first number system –

Unary number system

*

* *

* * * * *

* * * * * * * *   = 8   = 1000

* * * * * * * * *   = 9   = 1001

Number System

The first number system –

Inefficient – since we need n places (that hold digits) for storing n

Can we do better – can a single place (that holds a digit) be used to store multiple values ?

Number System

The first number system –

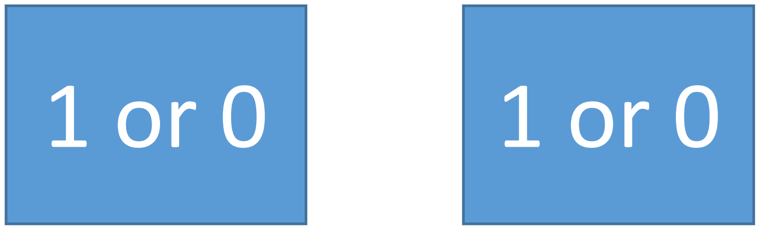
Inefficient – since we need n places (that hold digits) for storing n

Can we do better – can a single place (that holds a digit) be used to store multiple values ?

The next step – let one place store two possibilities – 0 or 1 OR even * or &

Number System

The number system where each digit place can have two options – called binary number system



Number System

The number system where each digit place can have two options – called binary number system

0 0

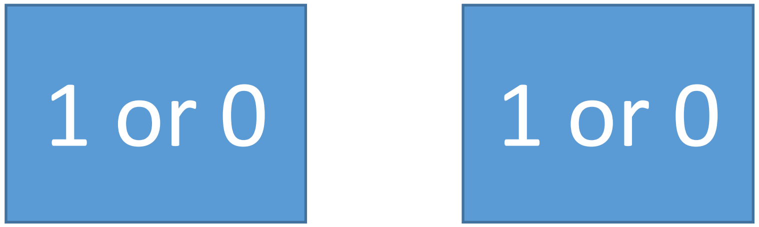
0 1

1 0

1 1

Total 4 possibilities –

What will be for 4 positions ?



Binary numbers using three bits -

Decimal		Binary		
0	→	0	0	0
1	→	0	0	1
2	→	0	1	0
3	→	0	1	1
4	→	1	0	0
5	→	1	0	1
6	→	1	1	0
7	→	1	1	1

Consider we have a number 1000 (decimal). To convert the number to binary – how many bits you need ?

2 bits you can represent upto	$3 = 2^2 - 1$
3 bits .	$7 = 2^3 - 1$
4	15
5 32	

X bits are there .... $2^X - 1 = 1000$

Number System

The number system where each digit place can have two options – called binary number system

What are the numbers for 8 positions ?

Binary Number	**Decimal equivalent**
0000 0000	0
….	1
….	…
1111 1111	255

Number System

How to convert any binary number directly to decimal number ?

What is a base – How many unique symbols we use in a certain number system.

Decimal number – has base 10 – 0,1,2,3,4,5,6,7,8,9

Binary number has base 2 – 0,1

Unary number has base 1 - 1

Convert a number in base x to base y.

Base 2 to Base 10

Binary to decimal

Number System

How to convert any binary number directly to decimal number ?

What is the decimal equivalent of the number ?

1	1	0	1	1	0	1	1

- How the number 1234 is composed in decimal number system ??

- $(1234)_{10} = 1000 * 1 + 100 * 2 + 10*3 + 4$

$= 10^3 * 1 + 10^2 *2 + 10^1 * 3 + 10^0 *4$

$(101)_2 = 2^2 * 1 + 2^1 * 0 \quad + 2^0 * 1 \qquad = 4 + 0 + 1 = 5$

Number System

How to convert any binary number directly to decimal number ?

Take a small example -

1	0	1

Number System

How to convert any binary number directly to decimal number ?

Take a small example -

1	**0**	**1**

= 5

Number System

How to convert any binary number directly to decimal number ?

Take a small example -

0	**1**	**1**

= 3

Number System

How to convert any binary number directly to decimal number ?

Take a small example -

1	1	1

= 7

Number System

How to convert any binary number directly to decimal number ?

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Number System

How to convert any binary number directly to decimal number ?

0	**0**	**0**	**0**
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

How to decompose a number

Decimal number 1234

= 1 X 1000 + 2 X 100 + 3 X 10 + 4

= 1 X 10^3 + 2X 10^2 + 3X10^1+ 4X10^0

= SUM (Terms)
Each Term = Coeff. X POWER of 10

Number System

How to convert any binary number directly to decimal number ?

Take a small example - 111

1	1	1
2^2	2^1	2^0

= 1X2^2 + 1 X 2^1 + 1 X 2^0 = 7

Number System

How to convert any binary number directly to decimal number ?

Take a small example -

0	1	1
2^2	2^1	2^0

= 3

Number System

How to convert any binary number directly to decimal number ?

What is the decimal equivalent of the number ?

1	1	0	1	1	0	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Sample conversion

1 0 0 0 0 1 0 0 0 0 0 0 0 1

For the 0th position = 1
For the 7th position its = 2^8 = 256
For the 12th position its = 2^13

Total value = 1 + 256 + 2^13

Number System

How to convert any binary number directly to decimal number ?

What is the decimal equivalent of the number ?

1	1	0	1	1	0	1	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	0	16	8	0	2	1

= 128 + 64 +16 + 8 + 2 + 1 = 219

Number System

How to convert any binary number directly to decimal number ?

What is the decimal equivalent of the number ?

1	0	1	1	0	1	1	1

= ?

Number System

How to convert any binary number directly to decimal number ?

What is the decimal equivalent of the number ?

1	0	1	1	0	1	1	1

= 183

Number System

How to convert any binary number directly to decimal number ?

What is the decimal equivalent of the number ?

1 0 0 1 0 0 1 0 0 1

Number System

How to convert any binary number directly to decimal number ?

What is the decimal equivalent of the number ?

1 0 0 1 0 0 1 0 0 1 = 585

Number System

MSB				Binary Digit				LSB
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
256	128	64	32	16	8	4	2	1

Number System

How to convert a decimal number directly to its equivalent binary number ?

What about 123 ?

$(123)_{10}$

Number System

Concept:

In binary we don’t have more than 2 symbols – so, we need to find how many times 2 are there ….

Strategy -

Go on dividing by 2

Go on collecting the remainders -

Number System

You have Z – a number

You want to represent using base-X number system

Z/X – divide the number z by x and get the remainder

Z is in decimal and X is in also decimal.

Number System

- **Repeated Division-by-2 Method**
- Divide-by-2 (two) to give a result and a remainder of either a “1” or a “0” until the final result equals zero.
- Example:

- Convert the decimal number 294_{10} into its binary number equivalent.

Number System

- **Repeated Division-by-2 Method**
- Divide-by-2 (two) to give a result and a remainder of either a “1” or a “0” until the final result equals zero.
- Example:

- Convert the decimal number 294_{10} into its binary number equivalent.

Number	**294**		
divide by 2			
result	147	remainder	**0** (LSB)
divide by 2			
result	73	remainder	**1**
divide by 2			
result	36	remainder	**1**
divide by 2			
result	18	remainder	**0**
divide by 2			
result	9	remainder	**0**
divide by 2			
result	4	remainder	**1**
divide by 2			
result	2	remainder	**0**
divide by 2			
result	1	remainder	**0**
divide by 2			
result	0	remainder	**1** (MSB)

Number System

294_{10}

is equivalent to

100100110_2

Number System

What is binary equivalent of 754

Number System

What is binary equivalent of 754

1011110010

Number System

What is binary equivalent of 1927

Number System

What is binary equivalent of 1927

11110000111

Number System

How many bits will be necessary to represent the following number to binary ?

177357827

Number System

How many bits will be necessary to represent the following number to binary ?

177357827
= 1010100100100100010000000011
(28 bits)

Number System

How many bits will be necessary to represent the following number to binary ?

Ceiling of Log (177357827) base 2 =
Ceiling of 27.4 = 28

int is enough

Finding number of bits ….

- Ceiling(8.0) = 8

- Ceiling(8.1) = 9

- If the value is 15 = how many bits you need ?
- Log(15) base 2 = 3.some fraction?
- Log (16) base 2 = 4 = Ceiling function + 1
- Log(17) base 2 = 4.some fraction = Ceiling function = 5
- Log 32 = 5.0 … - You need 6 bits ….

- 1555239234

Number System

Number of Bytes	Common Name
1,024 (2^{10})	Kilobyte (KB)
1,048,576 (2^{20})	Megabyte (MB)
1,073,741,824 (2^{30})	Gigabyte (GB)
a very long number! (2^{40})	Terabyte (TB)

Number System

Any numbering system can be summarized by the following relationship:

$$N = \mathbf{b_n\, q^n \ldots\, b_3\, q^3 + b_2\, q^2 + b_1\, q^1 + b_0\, q^0} +$$

$$\mathbf{b_{-1}\, q^{-1} + b_{-2}\, q^{-2} \ldots}$$

etc.

$N = b_i\, q^i$

where: N is a real positive number

b is the digit

q is the base value

and integer (i) can be positive, negative or zero

Number System

How to convert a binary fraction to decimal ?

Say 0.1011 –

what is the decimal equivalent ?

Number System

How to convert a binary fraction to decimal ?

This image cannot currently be displayed.

Say 0.1011 –

This image cannot currently be displayed.

Number System

What is equivalent to 1101.0111 in decimal ?

Number System

1101.0111

=

$(1\times2^{3}) + (1\times2^{2}) +$

$(0\times2^{1}) + (1\times2^{0}) +$

$(0\times2^{-1}) + (1\times2^{-2}) +$

$(1\times2^{-3}) + (1\times2^{-4})$

Number System

1101.0111

=

$(1\times2^{3}) + (1\times2^{2}) +$

$(0\times2^{1}) + (1\times2^{0}) +$

$(0\times2^{-1}) + (1\times2^{-2}) +$

$(1\times2^{-3}) + (1\times2^{-4})$

= 8 + 4 + 0 + 1 + 0 + 1/4 + 1/8 + 1/16

= 8 + 4 + 0 + 1 + 0 + 0.25 + 0.125 + 0.0625 = 13.4375_{10}

1101.0111_{2} is same as 13.4375_{10}

Number System

Evaluate -

0.11

11.001

1011.111

Number System

Evaluate -

0.11 = $(1\times2^{-1}) + (1\times2^{-2}) = 0.5 + 0.25 = \mathbf{0.75_{10}}$

11.001 =
$(1\times2^{1}) + (1\times2^{0}) + (1\times2^{-3}) = 2 + 1 + 0.125 = \mathbf{3.125_{10}}$

1011.111 =
$(1\times2^{3}) + (1\times2^{1}) + (1\times2^{0})\ (1\times2^{-1}) + (1\times2^{-2}) + (1\times2^{-3})$
$= 8 + 2 + 1 + 0.5 + 0.25 + 0.125 = \mathbf{11.875_{10}}$

Number System

How to convert a decimal fraction to binary fraction ?

Say 1234.567

Number System

Find the binary fraction equivalent of the decimal fraction: 0.8125_{10}

Number System

Find the binary fraction equivalent of the decimal fraction: 0.8125_{10}

0.8125 (multiply by 2)

= **1**.625 = 0.625 carry **1** (MSB)

0.625 (multiply by 2)

= **1**.25 = 0.25 carry **1** (↓)

0.25 (multiply by 2)

= **0**.50 = 0.5 carry **0** (↓)

0.5 (multiply by 2)

= **1**.00 = 0.0 carry **1** (LSB)

Number System

Find the binary fraction equivalent of the decimal fraction: 0.8125_{10}

0.8125 (multiply by 2)

= **1**.625 = 0.625 carry **1** (MSB)

0.625 (multiply by 2)

= **1**.25 = 0.25 carry **1** (↓)

0.25 (multiply by 2)

= **0**.50 = 0.5 carry **0** (↓)

0.5 (multiply by 2)

= **1**.00 = 0.0 carry **1** (LSB)

0.8125_{10} is equivalent to 0.1101_2 ← (LSB)

Number System

Cross check

0.1101

$= 0.5 + 0.25 + 0.0625 = \mathbf{0.8125_{10}}$

Number System

Example

Find the binary fraction equivalent of the following decimal number:

54.6875

Number System

Find the binary fraction equivalent of the following decimal number: 54.6875

54 (divide by 2) = 27 remainder **0** (LSB)
27 (divide by 2) = 13 remainder **1** (↑)
13 (divide by 2) = 6 remainder **1** (↑)
6 (divide by 2) = 3 remainder **0** (↑)
3 (divide by 2) = 1 remainder **1** (↑)
1 (divide by 2) = 0 remainder **1** (MSB)

Number System

Find the binary fraction equivalent of the following decimal number: 54.6875

binary equivalent of 54_{10} is therefore: 110110_2

Number System

Decimal fraction 0.6875 to a binary fraction -

0.6875 (multiply by 2)

= **1**.375 = 0.375 carry **1** (MSB)

0.375 (multiply by 2)

= **0**.75 = 0.75 carry **0** (↓)

0.75 (multiply by 2)

= **1**.50 = 0.5 carry **1** (↓)

0.5 (multiply by 2)

= **1**.00 = 0.0 carry **1** (LSB)

Number System

Binary equivalent of 0.6875_{10} is 0.1011_2 $\leftarrow$ (LSB)

Binary equivalent of the decimal number:

54.6875_{10} is 110110.1011_2

Special issues

```
void main()
{
    float a = 0.7;
    if (a < 0.7)
        printf("value :  %f",a);
    else if (a == 0.7)
        printf("equal values");
    else
        printf("hello");
}
```

What should be the output ?

Special issues

float a = 0.7

The variable a is float

0.7 is a double

Special issues

```
If(a < 0.7)

If (a == 0.7)
```

a is promoted to double and then compared

Special issues

In certain special cases binary fractions cannot be stored properly in finite number of bits in computer

Example:

Convert 0.5_{10} $\rightarrow$ 0.1

Special issues

In certain special cases binary fractions cannot be stored properly in finite number of bits in computer

But what about 0.1 ?

Special issues

Apply the same rule – (recap)

1. Multiply by two
2. Take decimal as the digit
3. Take the fraction as the starting point for the next step
4. Repeat until you either get to 0 or a periodic number
5. Read the number starting from the top - the first result is the first digit after the point

Special issues

0.1 * 2 = 0.2 -> 0
0.2 * 2 = 0.4 -> 0
0.4 * 2 = 0.8 -> 0
0.8 * 2 = 1.6 -> 1
0.6 * 2 = 1.2 -> 1
0.2 * 2 = 0.4 -> 0
0.4 * 2 = 0.8 -> 0
0.8 * 2 = 1.6 -> 1
0.6 * 2 = 1.2 -> 1

Result:
0.0001100110011001….

0.00011(0011) periodic.

Special issues

If you can store upto infinity – then the actual number can be stored.

If you store in limited number of bits – less than the actual number is stored

Example -

Special issues

Try to store 0.1 – in 8 bits

0.00011001

Special issues

Try to store 0.1 – in 8 bits

But 0.00011001 binary is not 0.1

Is actually - ?

Special issues

Try to store 0.1 – in 8 bits

But 0.00011001 binary is not 0.1

Is actually - ?

0.09765625 (less, close to 0.1)

Special issues

Can the value go higher than 0.1 ?

Special issues

Can the value go higher than 0.1 ?

Yes

This may happen due to rounding up ….

Special issues

Example:

in double – 53 bits are used for storing the number …

This rounds up the infinite representation upto 53 significant bits.

Special issues

0.1 in 57 bits in binary ….

0.0001100110011001100110011001100110011001100
11001100110011001…
*

Bits 54 and beyond total to greater than half the value of bit position 53, so this rounds up to

0.0001100110011001100110011001100110011001100
110011001101

Special issues

But -

0.00011001100110011001100110011001100110011001100110011001101

In decimal, this is -

0.1000000000000000055511151231257827021181583404541015625

which is slightly greater than 0.1.

Special issues

If you were to print that to 17 significant decimal digits you’d get 0.10000000000000001

(printing rounds the result as well).

Note that if you were to print to less than 17 digits, the answer would be 0.1.

That’s just an illusion **though — the computer has not stored 0.1.**

Special issues

It Can Be Slightly Greater or Slightly Less Than 0.1

Depending on how many bits of precision are used

The floating-point approximation of 0.1 could be *less* than 0.1.

For 11 significant bits, 0.1 rounds to 0.0001100110011 in binary, which is 0.0999755859375 in decimal.

Special issues

```
void main()
{
    float a = 0.7;
    if (a < 0.7)
        printf("value :  %f",a);
    else if (a == 0.7)
        printf("equal values");
    else
        printf("hello");
}
```

What should be the output ?

Special issues

0.7 **(double is larger than float)**

in single precision (float)
0.699999988079071044921875

In double precision (double)
0.6999999999999999555910790149937383
830547332763671875

Special issues

0.3 **(float is larger than double)**

In float

0.300000011920928955078125

In double

0.299999999999999988897769753748434595763683319091796875

Hex, Octal, Binary, Decimal, ….

Generalizations ….