

Simple Java program

```
import java.io.*;
```

```
class MYInteger{  
    private int value;  
    MYInteger(int i){  
        value = i;  
    }  
}
```

```
public class IntegerTest {  
    public static void main(String args [] ){  
        int integer1 =77;  
        Integer i1 = new Integer(integer1);  
        Integer i2 = new Integer(integer1);  
        System.out.println("\n i1==i2 " + (i1==i2) );  
        System.out.println(" i1.equals(i2) " + (i1.equals(i2)) );  
        System.out.println("\n The binary of " +integer1 +" is " +Integer.toBinaryString(integer1) );  
        System.out.println(" The Octalof " +integer1 +" is " +Integer.toOctalString(integer1) );  
        System.out.println(" The Hex of " +integer1 +" is " +Integer.toHexString(integer1) );  
    }  
}
```

Simple Java program 2

```
/* Con't */
    MYInteger mi1 = new MYInteger(integer1);
    MYInteger mi2 = new MYInteger(integer1);
    System.out.println(" \n mi1==mi2 " + (mi1==mi2) );
    System.out.println(" mi1.equals(mi2) " + (mi1.equals(mi2)) );
    /**/

int max32 = Integer.MAX_VALUE;
System.out.println("\n\n\n The binary of " + max32 + " is " + Integer.toBinaryString(max32) );
System.out.println(" \n The Octal of " + max32 + " is " + Integer.toOctalString(max32) );
System.out.println("\n The Hex of " + max32 + " is " + Integer.toHexString(max32) );

int min32 = Integer.MIN_VALUE;
System.out.println("\n\n\n The binary of " + min32 + " is " + Integer.toBinaryString(min32) );
System.out.println(" \n The Octal of " + min32 + " is " + Integer.toOctalString(min32) );
System.out.println("\n The Hex of " + min32 + " is " + Integer.toHexString(min32) );

}
}
```

Simple Java program 3

How to run it?

1. <http://java.sun.com/docs/books/tutorial/getStarted/cupojava/win32.html>
2. Run "Hello Java" Example
3. Then compile our java example by typing

```
javac TestInteger.java
```

4. To run it type

```
java TestInteger
```

Note the filename is
case sensitive!

Simple Java program 4

```
G:\MyData\CCNYCOURSES\cs342_Spring2007\Lec03_Feb_07_2007>javac IntegerTest.java
G:\MyData\CCNYCOURSES\cs342_Spring2007\Lec03_Feb_07_2007>java IntegerTest

i1==i2 false
i1.equals(i2) true

The binary representation of 77 is 1001101
The Octal representation of 77 is 115
The Hex representation of 77 is 4d

mi1==mi2 false
mi1.equals(mi2) false

The binary representation of 2147483647 is 11111111111111111111111111111111
The Octal representation of 2147483647 is 17777777777
The Hex representation of 2147483647 is 7fffffff

The binary representation of -2147483648 is 10000000000000000000000000000000
The Octal representation of -2147483648 is 20000000000
The Hex representation of -2147483648 is 80000000
```

Assignment No. 3

1.

In this exercise program we are interested to input from the keyboard 32, 64, 128, 256, 512, and 1024 bit integers and store their binary representations in memory.

Write a program that can input up to a 1000 digits from the keyboard.

To input each digit use `getc()` function(get character).

Store digits in memory.

Convert this multidigit number to a binary representation and then store it in memory.

Print out the numbers in binary.

2.

Write a program to compute

$$M^{\underline{Q} \cdot P} \bmod N$$

where M,P,Q,N are integers of sizes 32, 64, 128, 256, 512, 1024 bits.

Measure and print out computation times for each integer size.

Run the above program on two computers with different processors.

Compare the results.

3. The purpose of this program is to understand and experience the limits of standard data types when dealing with irrational numbers. Each standard data type has its min and max values. Thus the number of digits representing irrational number is limited(precision is limited).

Write a program that computes and stores in memory any specific irrational number(choose any number you like) with a desired degree of precision.

For example, you can compute Pi with 10, 20, 100, 200, 300, 400 digits after the decimal point.

Measure and report the computation time.

Encoding Information

Encoding describes the process of assigning representations to information

Fixed-length codes

4-BIT Binary Code Decimal(BCD)- represents decimal digits {0,1,2,3,4,5,6,7,8,9,0}

0101 ~ 5; 1001 ~ 9

8-BIT ASCII Code represents characters: {A-Z}, {a-z}, {1,2,3,4,5,6,7,8,9,0}, {symbols, signs}

0011 0101 ~ '5'; 0011 1001 ~ '9'

8-BIT Code may represent machine instruction operation: add, sub, mul, beq...

8-BIT Code may represent one of 256 colors of a pixel in a DIGITAL IMAGE (1024x1024 pixels)

16-BIT Code may represent an amplitude (one from 65k) of one DIGITAL SOUND sample

6 x 16-BIT = 96 bits Code may represent a sample of 6 channel DIGITAL SOUND

1 sec of digital sound ~ 16 bit x 40,000 samples = 640 000 bits /sec

Encoding Information

Binary Coded Decimal (BCD)

Fixed-length codes

4-BIT Binary Code Decimal(BCD)- represents decimal digits {0,1,2,3,4,5,6,7,8,9,0}

0101 ~ 5; 1001 ~ 9

1237 -> 0001 0010 0011 0111

16 bits are used to represent 1237

Encoding Information

Encoding describes the process of assigning representations to information

Fixed-length codes

8-BIT Code may represent machine instruction operation: add, sub, mul, beq...

For IA-32 instruction set summary see Figure 2.46 and 2.45...

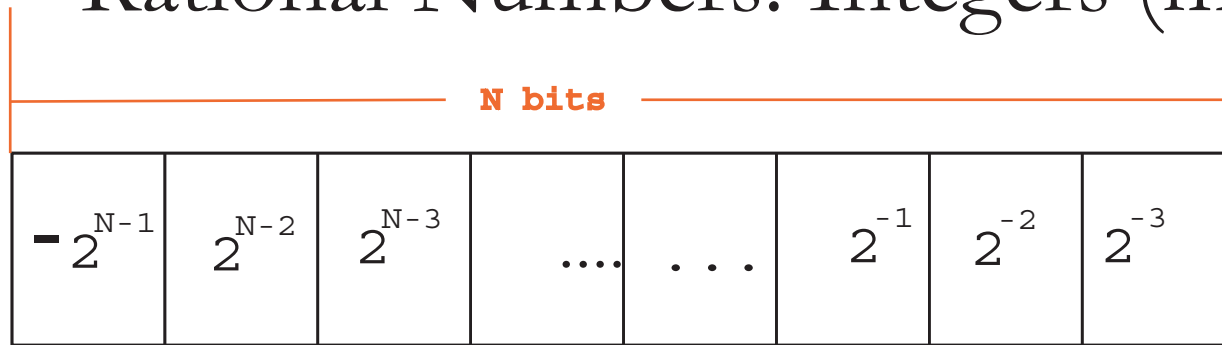
6-BIT Code is used in MIPS instruction set to encode machine instructions : add, sub, mul, beq...

For MIPS instruction set summary see Figure 2.20.

Encoding Information

Encoding numbers

Rational Numbers: Integers (m,n) and $n \neq 0$



Sign bit

Fixed point

Moving the location of fixed point we determine the integer range and fraction range.

$$11000101 = -2^4 + 2^3 + 2^{-1} + 2^{-3} = -16 + 8 + 0.5 + 0.125 = -7.325$$

Fixed point

$$11000101 = -2^1 + 2^0 + 2^{-4} + 2^{-6} = -2 + 1 + 0.125 + 0.03125 = -0.84375$$

Fixed point

Rational Number Program

Objective

Your assignment is to implement a program that will be capable of adding, subtracting, multiplying and dividing rational numbers.

Example

If you enter two rational numbers $\frac{1}{2}$ and $\frac{1}{2}$ you should get the following results.

$$\frac{1}{2} + \frac{1}{2} = \frac{2}{2} = 1$$

$$\frac{1}{2} - \frac{1}{2} = 0$$

$$\frac{1}{2} * \frac{1}{2} = \frac{1}{4}$$

$$\frac{1}{2} / \frac{1}{2} = \frac{2}{2} = 1$$

After typing the same rational numbers into the program we get results that should look similar to the results above. An example is shown below.

```

E:\PROGRA~1\XINOX5~1\JCREAT~1\GE2001.exe
1nd Rational Number
Input First Number: 1
Input Second Number: 2

2nd Rational Number
Input First Number: 1
Input Second Number: 2

-----Choose an Operation-----
To add rationals.....<Type 1>
To subtract rationals.<Type 2>
To multiply rationals.<Type 3>
To divide rationals...<Type 4>
1
<1 over 2> + <1 over 2> = <2 over 2>
Would like to do another Operation (y or n): y

-----Choose an Operation-----
To add rationals.....<Type 1>
To subtract rationals.<Type 2>
To multiply rationals.<Type 3>
To divide rationals...<Type 4>
2
<1 over 2> - <1 over 2> = 0
Would like to do another Operation (y or n): y

-----Choose an Operation-----
To add rationals.....<Type 1>
To subtract rationals.<Type 2>
To multiply rationals.<Type 3>
To divide rationals...<Type 4>
3
<1 over 2> * <1 over 2> = <1 over 4>
Would like to do another Operation (y or n): y

-----Choose an Operation-----
To add rationals.....<Type 1>
To subtract rationals.<Type 2>
To multiply rationals.<Type 3>
To divide rationals...<Type 4>
4
<1 over 2> / <1 over 2> = <2 over 2>
Would like to do another Operation (y or n): n

```

Now let's try another set of rational numbers to check if the program truly works for this

lets choose the rational numbers $\frac{1}{3}$ and $\frac{1}{2}$. Here the results should be.

$$\frac{1}{3} + \frac{1}{2} = \frac{5}{6}$$

$$\frac{1}{3} - \frac{1}{2} = -\frac{1}{6}$$

$$\frac{1}{3} * \frac{1}{2} = \frac{1}{6}$$

$$\frac{1}{3} / \frac{1}{2} = \frac{2}{3}$$

After typing these rational numbers into the program we get results that should look similar to the results above.

```
c:\ E:\PROGRA~1\XINOXS~1\JCREAT~1\GE2001.exe
Would like to enter another Rational (y or n): y

1nd Rational Number
Input First Number: 1
Input Second Number: 3

2nd Rational Number
Input First Number: 1
Input Second Number: 2

-----Choose an Operation-----
To add rationals.....(Type 1)
To subtract rationals.(Type 2)
To multiply rationals.(Type 3)
To divide rationals...(Type 4)
1
<1 over 3> + <1 over 2> = <5 over 6>
Would like to do another Operation (y or n): y

-----Choose an Operation-----
To add rationals.....(Type 1)
To subtract rationals.(Type 2)
To multiply rationals.(Type 3)
To divide rationals...(Type 4)
2
<1 over 3> - <1 over 2> = <-1 over 6>
Would like to do another Operation (y or n): y

-----Choose an Operation-----
To add rationals.....(Type 1)
To subtract rationals.(Type 2)
To multiply rationals.(Type 3)
To divide rationals...(Type 4)
3
<1 over 3> * <1 over 2> = <1 over 6>
Would like to do another Operation (y or n): y

-----Choose an Operation-----
To add rationals.....(Type 1)
To subtract rationals.(Type 2)
To multiply rationals.(Type 3)
To divide rationals...(Type 4)
4
<1 over 3> / <1 over 2> = <2 over 3>
Would like to do another Operation (y or n): n
Would like to enter another Rational (y or n): n
Press any key to continue...
```

Encoding Information

Encoding describes the process of assigning representations to information

Digital Sound

8-BIT may encode one amplitude (level) out of 256 per one sample.

16-BIT Code may represent an amplitude (one from 65k) of one DIGITAL SOUND sample

SAMPLING RATE = Number of samples per one second.

6 x 16-BIT = 96 bits may encode one sample of a 6 channel DIGITAL SOUND

Number of bits required to encode 1 sec of digital sound =
(Number of bits per sample) x (Number of samples per one sec)

How much memory do you need to store 1 hour of stereo music using 16 bit quantization and sampling rate =40,000 samples/sec ?.

Encoding Information

Encoding describes the process of assigning representations to information

Digital Images, Video

1-BIT can be used to encode "BLACK" or "WHITE" color.

e.g. an image of size 1024x1024 pixels (points) may be encoded using 1 bit per pixel ("1"~ black, "0"~ white). Such image size is - 1Megabit.

8-BIT Code may represent one of 256 GRAY LEVELS between black and white for each pixel in a DIGITAL IMAGE (1024x1024 pixels). Such image size is - **1MegaByte**.

32 -BIT Code may encode true color of a pixel in a DIGITAL IMAGE (1024x1024 pixels). Such image size is - 4 **MegaBytes**.

Digital video is 60 Frames/ sec.

1 Frame of a true color digital image of size (1024x1024 pixels) requires 4 MegaBytes to store it.
Video Throughput: **240 MegaBytes/sec**.

How much memory do you need to store 1 HOUR of digital video?