Birla Institute of Technology & Science, Pilani

Software Development & Educational Technology Unit

First Semester, 2007-2008

SDETC102 Core Java

Assignment 1

Submission: Before/At 20th November 2007, 5:00 PM via. Intrabits

Base64 is a scheme used for encoding email attachments and sending them across the internet. The purpose of this assignment is to give the student a fair idea about issues involved in processing streams of data.

1. Implement either of these classes:

```
public class Base64Encoder {

      /* Returns the number of bytes output after encoding.
      The length is always a multiple of 4. */

      public int encodeBytes(byte in[], int inputlen, byte
      out[]);

}
```

or:

```
public class Base64Decoder {

      /* Returns the number of bytes output after decoding.
      The input length is always a multiple of 4. The output
      length has no restriction. */

      public int decodeBytes(byte in[], int inputlen, byte
      out[]);

}
```

A description of the algorithm used:

Encoding: Consider an input stream of length n characters; Let us assume now a byte size is *8 bits* only. So, we have a total of *n*8* bits in our input stream. Now, we have to split it into chunks of maximum size *6 bits* each. So, we will have *(n*8)/6* full chunks of *6*

bits each and optionally, a trailing chunk of size *(n*8)%6* bits, which we will pad up with as many bits required to form a full *6* bit chunk (*2* or *4*). We always pad with *0s (zeroes)*.

Now, with each obtained chunk, look at each chunk representing a *6* bit integer *n*. Now we retrieve the *nth* character of this string/array (of length *64* bytes):

```
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/
```

and we output that character.

If we have not padded any bits, the algorithm ends here.

If we had padded *4* bits to the last chunk, we additionally output two == characters after we output the encoded last chunk. If we had padded 2 bits to the last chunk, we additionally output a single = character.

This makes the total number of characters we output a multiple of *4*, making decoding simpler, negating the need to send the length of the encoded stream.

Decoding: Consider an input stream of length *n* characters; The length of the stream is a multiple of *4* bytes.

Decode the *4* bytes into *3* bytes as follow: Find the number representing the *1st* character in the string array we used in the encoding process. Now take this as a *6* bit number, do so for the rest

a. To the *1st* *6* bit number, concatenate the first *2* bits of the *2nd* number, and output it.

b. To the last *4* bits of the *2nd* number, concatenate *4* bits of the *3rd* number and output it.

c. Now concatenate the last *2* bits of *3rd* number to all bits of the last number and output it.

If the *3rd* and *4th* byte of the last *4* byte chunk are =, then you only need to do step *a*. If only the *4th* byte is =, do steps *a* and *b*.

This For each byte you output, add it to the count of output characters and return it. helps in determining how much of the output buffer you should write.

2. Now implement either of these classes:

```
public class Base64EncodeDriver {

public static void main(String []args){}

}
```

```
public class Base64DecodeDriver {

public static void main(String []args){}

}
```

and perform appropriate encoding or decoding, using the class written in problem 1. Use *6144* as the buffer size for encoder and *8192* as the buffer size for decoder. Assume *input.txt* contains the plain/encoded input and *output.txt* will contain the encoded/plain output.

Sample plain text for encoding/decoding verification (*23* bytes):

```
This is the test file.
```

Encoded text for decoding/encoding verification (*32* bytes):

```
VGhpcyBpcyB0aGUgdGVzdCBmaWxlLgo=
```

Mode of submission: Through the Intrabits site. This assignment is also uploaded there.

Submission Guidelines: Create the driver and encoder/decoder file. Put them along with a text file giving details about limitations on implementation (if any) and your name and ID number. The zip file should be named *YEARDEGRXXX.zip* (eg: *2001HS12832.zip*) and submitted prior to the deadline mentioned. If any errors during submission, email the assignment to the course instructor.

Strict Warning against Plagiarism/Code Use: Implement all modules on your own. If you require help understanding the algorithm, discussions on that are permitted. Re-use of someone else's code on the Internet or otherwise is strictly condemned and will jeopardize your evaluation.