Problem 6.1:

a)

Simplicity, carried to the extreme, becomes elegance.

- Jon Franklin

b)

UTF-8	Unicode	Name
7c	U+007C	VERTICAL LINE
7b	U+007B	LEFT CURLY BRACKET
2d	U+002D	HYPHEN-MINUS
7d	U+007D	RIGHT CURLY BRACKET
e2 88 aa	U+222A	UNION
7b	U+007B	LEFT CURLY BRACKET
e2 88 92	U+2212	MINUS SIGN
7d	U+007D	RIGHT CURLY BRACKET
e2 88 aa	U+222A	UNION
7b	U+007B	LEFT CURLY BRACKET
c2 ad	U+00AD	SOFT HYPHEN
7d	U+007D	RIGHT CURLY BRACKET
e2 88 aa	U+222A	UNION
7b	U+007B	LEFT CURLY BRACKET
e2 80 91	U+2211	N-ARY SUMMATION
7d	U+007D	RIGHT CURLY BRACKET
e2 88 aa	U+222A	UNION
7b	U+007B	LEFT CURLY BRACKET
e2 80 94	U+2014	EM DASH
7d	U+007D	RIGHT CURLY BRACKET
e2 88 aa	U+222A	UNION
7b	U+007B	LEFT CURLY BRACKET
e2 80 94	U+2014	EM DASH
7d	U+007D	RIGHT CURLY BRACKET
7c	U+007C	VERTICAL LINE
20	U+0020	SPACE
3d	U+003D	EQUALS SIGN
20	U+0020	SPACE
37	U+0037	DIGIT SEVEN
0a	U+000A	<control>

c) UTF-32 needs 800000*4 = 3200000bytes
UTF-8 needs 800000*3 = 2400000bytes

Problem 6.2:

a)

No.3 and No.4 are equivalent.

These two have the same date: 2019-10-13, and the time of No.3 is 13:15:00 add 0 hours and 0 minutes.

No.4's time is 15:15:00, however, it takes away 2 hours which makes it equivalent with No.3.

b)

The "year 2038" problem is a time error problem which will be taking place on 19. Jan. 2038, at 3:14:07 am UTC.

Since most programs use Unix-like 32-bit operators for counting and storing time, which has the maximum decimal number of 2147483647,

which is 2^31 -1. This represents that in the binary system the storage will be full. After that moment, time will be wrapped around and the inner number will

become negative, therefore programs wouldn't be able to recognize the time as 2038.

There is not a perfect and universal solution for the problem, because every one of them will have some extra minor problems to deal with.

Anyways, the problem can be solved by changing the format of using int to unsigned int, which makes time storage be doubled, all the way to year 2106.

Another solution is to replace or update the programs with 64-bits so that storage will not be the case, however it is very difficult to make every program

into 64-bits, not to mention there are many devices cannot be updated via internet.

c)

The usage of offsetting is to determine local time with RFC time. While RFC 3339 allows an offset such as -00:00, it shows the local time has exactly time in hours and minutes, also the minus sign shows the local time has seconds away from the actual RFC time.