Ahmad Aziz
13220034

INTEGER PROBLEMS

a. Consider word size w = 4 bits, Fill in the following table indicating representation of $UMax_w$, $TMax_w$, $TMin_w$, -1 dan 0 in hexadecimal and give its numeric value.

$2^4 - 1 = 15$
$2^3 - 1 = 7$
$-2^3 = -8$

	Hexadecimal	Numeric value (decimal)
$Umax_4$	0xF	15
$Tmax_4$	0x9	7
$Tmin_4$	0x8	-8
-1	0x1	-1
0	0x0	0

b. Suppose that x and y have byte value. Fill in the following table indicating the byte values of the different C expression. Also give its unsigned and two's complement representations (in decimal)

y&x = 0100

0110

x (binary)	y (binary)	Operation	Binary	Unsigned	Two's comp
1010	0101	(x+y)	1111	15	-1
1111	0100	(y&x) + ~x	0100	4	4
1001	1000	(x^x) | y	1000	8	-8
1000	0000	x && (y-x)	0000	0	0
0110	1001	x & !y	0110	6	6
0110	0011	x && ~y	0000	0	0

c. Consider the following C functions:

```
int fun1 (unsigned word) {
   return (int) ((word << 27) >> 27);
}

int fun2 (unsigned word) {
   return ((int) word << 27) >> 27;
}
```

Assume these are executed on a machine with a 32-bit word size that uses two's-complement arithmetic. Assume also that right shifts of signed values are performed arithmetically, while right shifts of unsigned values are performed logically. Fill in the following table showing the effect of these functions for several example arguments. You will find it more convenient to work with a hexadecimal representation. Just remember that hex digit 8 through F have their most significant bit equal to 1.

w	fun1(w)	fun2(w)
15	0xF	0xF
16	0x10	0xFFFFFFF0
31	0x1F	0xFFFFFFFF
32	0x0	0x0