EXAMPLES:

1. CONVERTING DECIMAL TO BINARY

		Remainder
56	28	0
28	14	0
14	7	0
7	3	1
3	1	1
1	0	1

Divide by 2 at each step, storing the remainder. This gives the binary representation of the number.

$56_{10} = 111000_2$

2. CONVERTING DECIMAL TO HEX

		Remainder
126	7	14 (E)
7	0	7 (7)

Divide by 16 at each steps similar as above.

$126_{10} = 7E_{16} = \underset{7}{0111}\ \underset{E}{1110}_2$

3. TWO'S COMPLEMENT

Take the One's complement of the number and add 1

$126_{10} = 0111\ 1110_2$

1's complement = 1000 0001.

2's complement = 1000 0001 + 1. = 1000 0010

$X = 11100110_2$.

$-X = 00011010_2 = 26_{10}$

$X = -26_{10}$

Scanned by CamScanner

4. TYPECASTING:

a) Given a 5 bit register:

```
int a = -7
unsigned b = 5
c = a+b
```

$-a = 7 = 00111_2$ $\qquad b = 5 = 00101_2$

$a = 11001_2$

Signed int are typecast as unsigned integers giving us:

$$\begin{array}{rr} c = a+b = & 1\ 1\ 0\ 0\ 1 \\ + & 0\ 0\ 1\ 0\ 1 \\ \hline & 1\ 1\ 1\ 1\ 0_2 = 30_{10} \end{array}$$

b) Given a 6-bit register.

```
unsigned a = -12
int b      = -13
c = a + b
```

c) Given a 6-bit register.

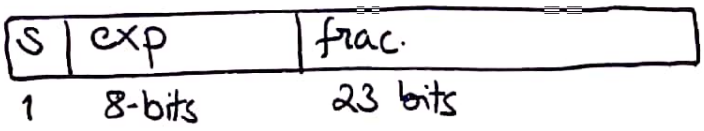
```
int a = -12.
int b = -21
c = a+b
printf("%u", c)
```

Scanned by CamScanner

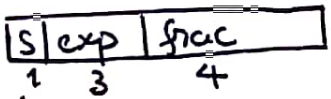
d) Given a 6-bit int and 12-bit long.

```
unsigned long a = 21
int b = -24
c = a + b
```

5. FLOATING POINTS:



a. Convert 2.625 to 8-bit floating point format.



A. Integral part, $2_{10} = 10_2$, for fractional part:

$0.625 \times 2 = 1.25$ [1]
$0.25 \times 2 = 0.5$ [0]
$0.5 \times 2 = 1.0$ [1]

$0.625_{10} = 0.101_2$

So, $2.625_{10} = 10.101_2 = 10.101_2 \times 2^0$

B. Normalize: $10.101_2 \times 2^0 = 1.0101_2 \times 2^1$

C. Mantissa : 0101

D. Exponent: $1 + 3 = 4 = 100_2$ → bias = $2^{k-1} - 1$

E. Sign bit is 0

Result | 0 | 1 0 0 | 0 1 0 1 |

b. Convert -4.75 to the 8-bit fp format.

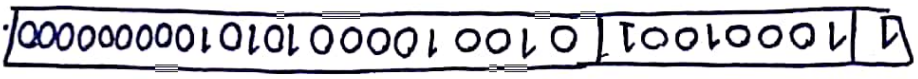
Ans. [illegible]

c. Convert 0.4625 to the 8-bit fp format

Ans. [illegible]

Scanned by CamScanner

d) Convert -1313.3125 to IEEE 32-bit FP.

Ans. 

Scanned by CamScanner