

1. (A) word : 32 bits
 (B) char : 8 bits
 (C) double : 64 bits
 (D) ~~2³²~~
 (E) ISA (Instruction Set Architecture)
 (F) input, output, control, memory, datapath

2. (A) F ex. move
 (B) F
 (C) T
 (D) F
 (E) T
 (F) T PC+4

3. R-format :

op	rs	rt	rd	shamt	funct
6	5	5	5	5	6
- I-format :

op	rs	rt	addr/immediate
6	5	5	16
- J-format :

op	address
6	26

4.

$$\frac{1}{(1 + 4 \times \frac{0.4}{2})^2} \times 100\% = 30.86\%$$

$$\frac{1}{(1 + 2 \times \frac{0.4}{2})^2} \times 100\% = 51.02\%$$

5. (A)

```
int yuck(int n)
{
    int i=1;
    int j=1;
    while(n >= 2)
    {
        k=i+j;
        i=j;
        j=k;
        n=n-1;
    }
    return j;
}
```

(B)

```
$a0 = 4 > 2
$t2 = 2
$t0 = 1
$t1 = 2
$a0 = 3

$a0 = 3 > 2
$t2 = 3
$t0 = 2
$t1 = 3
$a0 = 2
```

```
$a0 = 2 = 2
$t2 = 5
$t0 = 3
$t1 = 5
$a0 = 1

$a0 = 1 < 2
return $t1 = 5
```

Java C

可攜性好

沒有 assembler

6. (A)

2	28410	
2	14205	0
2	7102	1
2	3551	0
2	1775	1
2	887	1
2	443	1
2	221	1
2	110	1
2	55	0
2	27	1
2	13	1
2	6	1
2	3	0
2	1	1
	1	

$$28410_{10} = 110, 1110, 1111, 1010_2$$

$$-28410_{10} = 1111, 1111, 1111, 1111, 1001, 0001, 0000, 0110_2$$

(B) $0xCA50 = 1100, 1010, 0101, 0000_2$

(C) $0xFFF7 = 1111, 1111, 1111, 0111_2$

$$\rightarrow 0xFFF9 = 0000, 0000, 0000, 1001_2$$

$$0xFFF7 = -9_{10}$$

(D) $0x001F = 0000, 0000, 0001, 1111_2 = 31_{10}$

$$0xFFF7 = 1111, 1111, 1111, 0111_2 = -9_{10}$$

hex: $0xFFF7 - 0x001F = 0xFFD8$

dec: $-9 - 31 = -40$

E16
~~001F~~
 FF D8

7. (A)

addi	\$19, \$0, 0x20
lw	\$17, 0x04(\$19)
add	\$20, \$19, \$16
sw	\$20, 0x08(\$19)

$$\$19 = 0x00 + 0x20 = 0x20$$

$$\$17 = (0x20 + 0x04)_{\text{memory}} = 0x30$$

$$\$20 = 0x20 + 0x10 = 0x30$$

$$(0x20 + 0x08)_{\text{memory}} = 0x30$$

(B) addi \$1, \$0, 0x20

$$\$1 = 0x00 + 0x20 = 0x20$$

lw \$2, 0x04(\$3)

$$\$2 = (0x28 + 0x04)_{\text{memory}} = 0x50$$

add \$0, \$3, \$1

$$\$0 = 0x28 + 0x20 = 0x48$$

bne \$0, \$1, loop

$$0x48 \neq 0x20, \text{ jump to loop}$$

(C) Yes

8. bne \$to, \$zero, loop

\$to is nonzero \rightarrow jump to loop