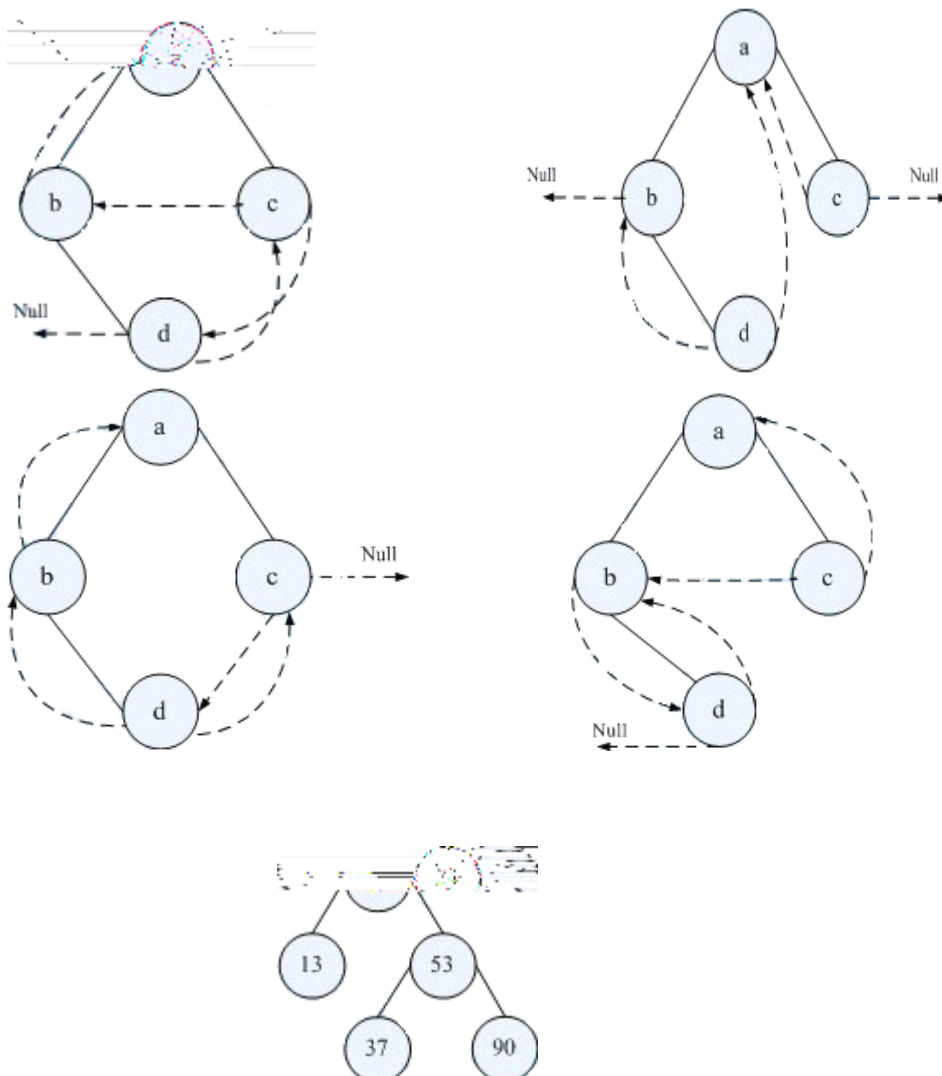
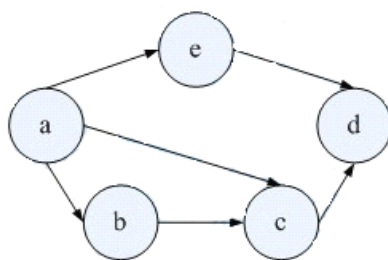
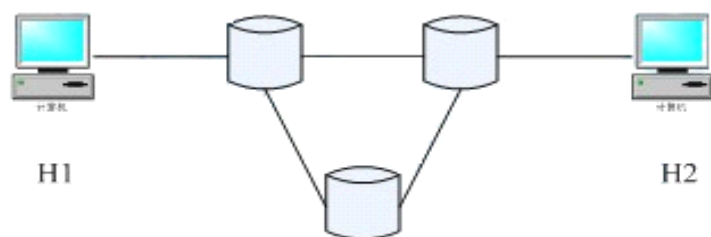


(1-40 , 2 , 80 , , , .)



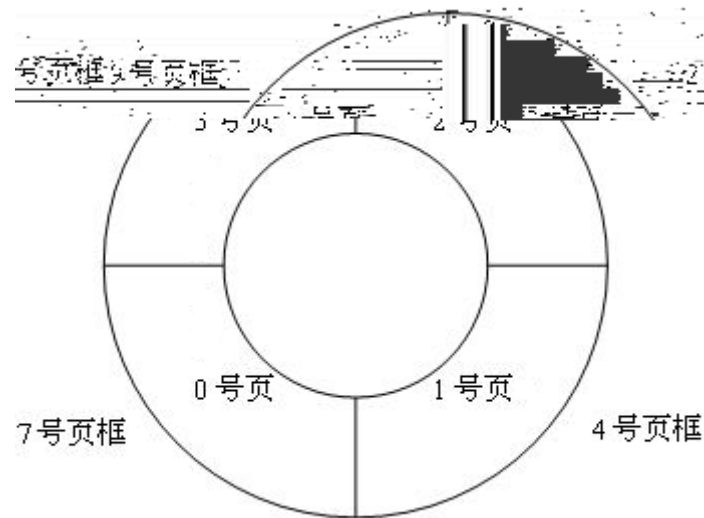


--	--	--



--	--	--	--	--

--	--



1.D

- A. Push, Push, Push, Push, Pop, Pop, Push, Pop, Pop, Push, Pop, Pop;
 B. Push, Push, Push, Pop, Pop, Push, Pop, Pop, Push, Pop, Push, Pop;
 C. Push, Push, Pop, Push, Pop, Pop, Push, Push, Pop, Push, Pop, Pop;
 D. Push, Pop, Push, Push, Push, Push, Push, Pop, Pop, Pop, Pop, Pop;

D

2.C

a b

C

a b

C

L

R

- A. aL(aR), bL, cR, dR, eR
 B. aL(aR), bL, cR, dL, eR
 C.
 D. aL(aR), bL, cR, dR, eL

3.D

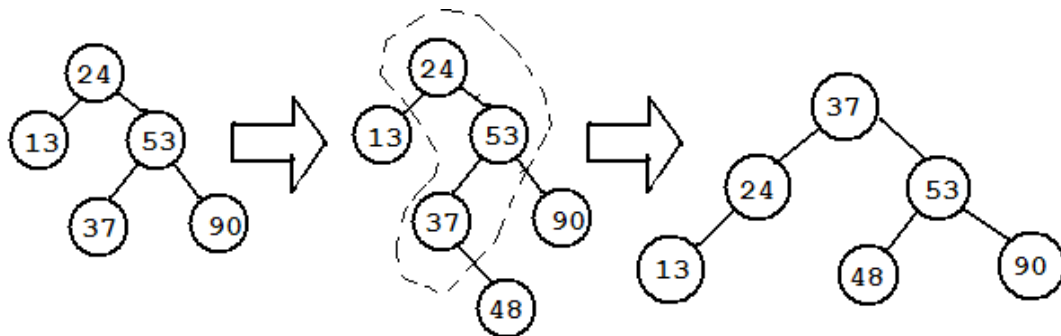
d, b, c, a d b
 b d c b
 a D

4.C

48

-1

-2



5.B

i i=0 1 2 3 4 N_i N
 $N-1$ $N=1+N_1+2N_2+3N_3+4N_4=N_0+N_1+N_2+N_3+N_4$
 $N_0 = 82$ T 82

6.A



7.C



8.B



9.B



10.D

A
B,C,D

```
void quicksort(int R[],int l,int r)
{
    ... ..
    quicksort(R,l,i-1);//
    quicksort(R,i+1,r);//
}
```

l r l r

$$F(l, r)$$

$$F(l, r) = F(l, i-1) + F(i+1, r) \dots\dots\dots$$

$$F(l, r) = F(i+1, r) + F(l, i-1) \dots\dots\dots$$

11.A

12.D

$$\text{CPI} \times T = *$$

13.B

8

8 -128 ~ +127

$$\begin{aligned} r1 &= -FGH - FEH = -2 \\ r2 &= -FGH - F2H = -14 \\ r3 &= -G0H - 90H = -70H \\ r4 &= -FGH - F8H = -8 \end{aligned}$$

$$G \quad 15 \quad r2 * r3$$

14.B

int float double

IEEE754 float double 7 15

d+f double 15 d+f f
0 d-d=0

15.D

2K×4 8K×8 8 2K×4 4 2 2K×4 2K×8
2 2K×4
0000 0000 0000 0000
0000 0111 1111 1111

0000 1000 0000 0000
0000 1111 1111 1111

0001 0000 0000 0000
0001 0111 1111 1111

0001 1000 0000 0000
0001 1111 1111 1111
0B1FH 0000 1011 0001 1111
0000 1000 0000 0000 0800H

16.A

RAM
Static RAM, SRAM)
Dynamic RAM, DRAM)
ROM

RAM RAM I
RAM ROM
Cache ROM Cache
RAM

17.D

Cache TLB
Cache TLB D Cache
Page

18.A

IR MAR MDR CPU
PC

CPU

19.A

20.D

A CRT
D

CPI

RAM

21.B

5 6 1 2 3 4
7 8 1~3

22.D

$$\frac{1600 \times 1200}{85} + 24$$
$$\text{BAND} = 1600 \times 1200 \times 24 \times 85 / 50\% = 7,833,600,000 \quad 7834\text{Mbps}$$

23.A

A

24.C

30.C

$$\begin{aligned}
 & 256/4=64 \\
 & 4 \times 256 = 1\text{KB} \\
 & 2 \times 64 \times 256 = 32\text{KB} \\
 & 64 \times 64 \times 256 = 1024\text{KB} \\
 & 1\text{KB} + 32\text{KB} + 1024\text{KB} = 1057\text{KB}.
 \end{aligned}$$

31.C

A B

32.B

B

33.C

7 5
C

34.C

$$\begin{aligned}
 & 980\text{B} \quad 20\text{B} \quad 980000 \quad 1000\text{B}, \quad 20\text{B} \quad 1000\text{B} \\
 & 1000000\text{B} \quad 1000 \quad 1\text{MB} \\
 & 100\text{Mbps} \quad 12.5\text{ MBps} \quad 100\text{Mbps}/8 \\
 & 1000000\text{B}/12.5\text{ MBps}=80\text{ms} \\
 & \text{H1} \quad \text{H1} \quad \text{H2} \quad \text{H1} \quad 2 \\
 & \text{H2} \quad 2*1000\text{B}/12.5\text{ MBps}=0.16\text{ms} \\
 & 80\text{ms}+0.16\text{ms}=80.16\text{ms} \quad \text{C}
 \end{aligned}$$

35.D

16 RIP 15
D

36.C

C

37.B

192.168.5.0/24
32 CIDR 2³-2=6
11111111 11111111 11111111 11111000 IP
8 5
0 1 3
0 1 B

38.D

D

39.A

212 =Min
=Min 4000 2000 =2000
2
2000-1000=1000 A

40.A

DNS IP

=
IP A

41

1 0.7 7
L = 7/0.7 = 10
T=10
H(key) = (key 3) MOD 10
H_i(key) = (H(key) + d_i) MOD 10 , (d_i = 1,2,3...9)

$H(7) = (7 - 3) \text{ MOD } 10 = 4$
 $H(8) = (8 - 3) \text{ MOD } 10 = 5$
 $H(30) = (30 - 3) \text{ MOD } 10 = 7$
 $H(11) = (11 - 3) \text{ MOD } 10 = 8$
 $H(18) = (18 - 3) \text{ MOD } 10 = 5$
 $H_1(18) = (H(18)+1) \text{ MOD } 10 = 6$
 $H(9) = (9 - 3) \text{ MOD } 10 = 6$
 $H(14) = (14 - 3) \text{ MOD } 10 = 1$

Hash

	0	1	2	3	4	5	6	7	8	9
	30	7	17	11	8	18		9		

2

$$ASL_1 = (1+1+1+1+2+1+1) / 7 = 8/7$$

$$ASL_2 = (7+6+5+4+3+2+1+2+1+1) = 3.2$$

42

```

1          p          R          p
    R          n-p          p
R          n-p
2          c
void Shift(int R[],int n,int p)//n          p
{
    int temp[p];    //
    int i=0;
    while(i<p)
    {
        // R          p
        temp[i]=R[i];
        i++;
    }
    i = 0;
    while(i<n-p) // R          p
    {
        R[i]=R[p+i];
        i++;
    }
    i = 0;
    while(i<p) //          p          R          n-p
    {
        R[n-p+i]=temp[i];
        i++;
    }
}
3          O(n)          O(p)

```

43

1 4 24=16
6 3 3
23=8
16 MDR 16
128KB 16 0~64K MA
R 16 216=64K
15 1
5+63+64
2 16 0~216-1
216 64K
0~216-1.

3 add (R4) , (R5) +

OP	Ms	Rs	Md	Rd
0010	001	100	010	101

0010 0011 0001 0101B=2315H

R4

R5

R5

R4 1234H R4 unneir 5678H
R5 5678H 1234H
R5 5678H+1=5679H

1234H

R5=5679H

R5 5678H =68ACH

1234

1234

44

1 cache

		TAG	Data

Block

cache

4

V 1

Cache

TAG

Cache

TAG Data

:

TAG		
-----	--	--

28 228=258M

6 26=64 5~0

3 23=8 8~6

TAG

TAG

Cache

2

a[0][31]

320+31*4=444=0001 1011 1100B

a[1][1]

320+ 256+1 *4=1348=0101 0100 0100B

8~6

6 110B 5 101B

3 a

64B

16

A,

Cache

15

Cache

16

216-212 /216 *100%=93.75%.

B

a

Cache

0.

a

1KB>64B

0

256

Cache

8

B

0.

Cache

A

Cache

B

1 V TAG

Cache

2

3

B A A B
 A A B B
 2
 82 11
 2km, 200000km/s
 $=2\text{km}/200000\text{km/s}=0.01\text{ms}$
 , 0.01ms 0.02ms
 2 A =1518 8b/10 Mb/s=1.2144ms
 64
 $=64 \quad 8\text{b}/10 \text{ Mb/s}=0.0512\text{ms}$
 0.02ms
 $=1.2144\text{ms}+0.0512\text{ms}+0.02\text{ms}=1.2856\text{ms}$
 1.2856ms 1518B
 1518B 18 90
 1500
 $=1500 \quad 8\text{b}/1.2856\text{ms} \quad 9.33\text{Mb/s}$