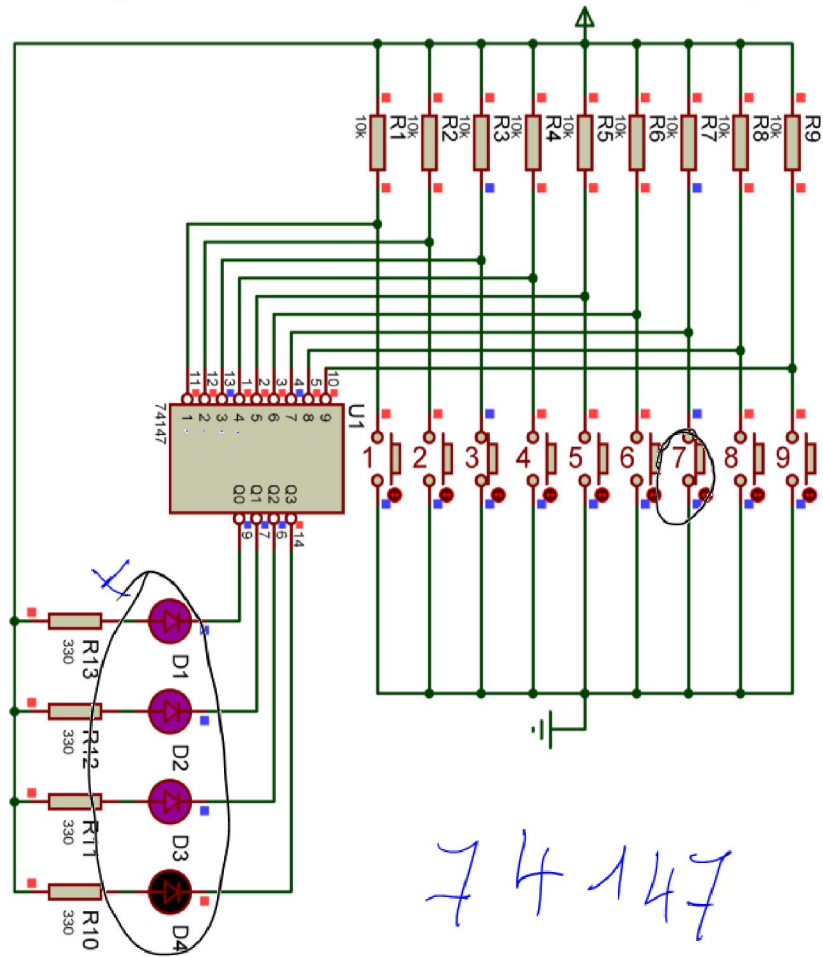


Exercice 1 : Fonction codage décimal/BCD

1- Donner les noms des entrées et des sorties du circuit

74147 ? 1 2 3 4 5 6 7 8 9 // 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

2- Compléter le tableau en se référant au montage



base décimale
0 → 9

74147

1001

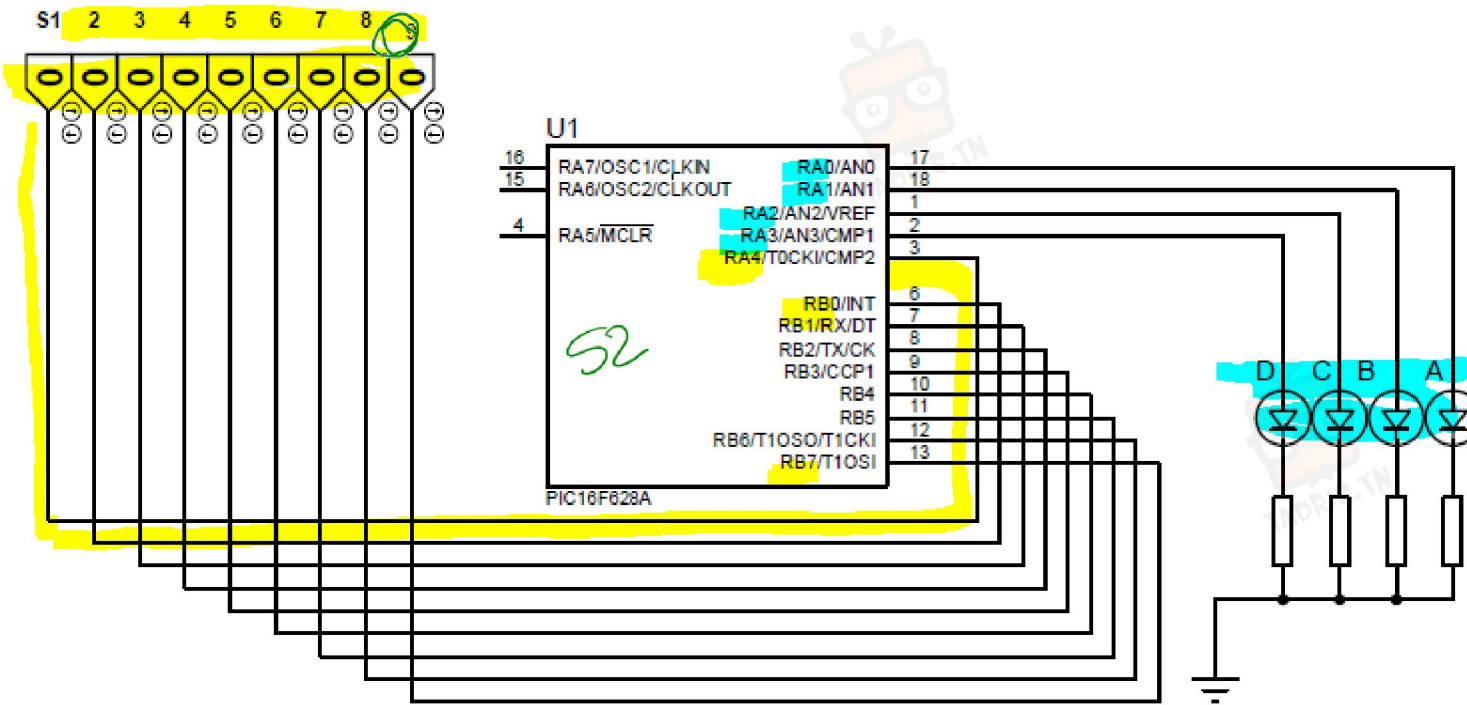
4 bits BCD

9	8	7	6	5	4	3	2	1	D	C	B	A
0	0	0	0	0	0	0	0	0	0	0	0	0
1	x	x	x	x	x	x	x	x	1	0	0	1
0	1	x	x	x	x	x	x	x	1	0	0	0
0	0	1	x	x	x	x	x	x	0	1	1	1
0	0	0	1	x	x	x	x	x	0	1	1	0
0	0	0	0	1	x	x	x	x	0	1	0	1
0	0	0	0	0	1	x	x	x	0	1	0	0
0	0	0	0	0	0	1	x	x	0	0	1	1
0	0	0	0	0	0	0	1	x	0	0	1	0
0	0	0	0	0	0	0	0	1	0	0	0	1

x : 0 ou 1

3- Si deux entées sont activées en même temps que se passe il ?
la plus grande valeur sera affichée.

4- Solution programmée



Compléter le programme en mikroC

Programme

```

sbit S1 at RA4_bit;
char S2;
void main()
{
    trisA = 0b...1...0; 0 0 0;

```

```

if (portb.b7) S2 =

```

```

si RB7=1 S2 =
portA.F4

```

```

trisB = 0xFF;
CMCON = 0x07;

```

```

S2 = 0;
while(1)

```

```

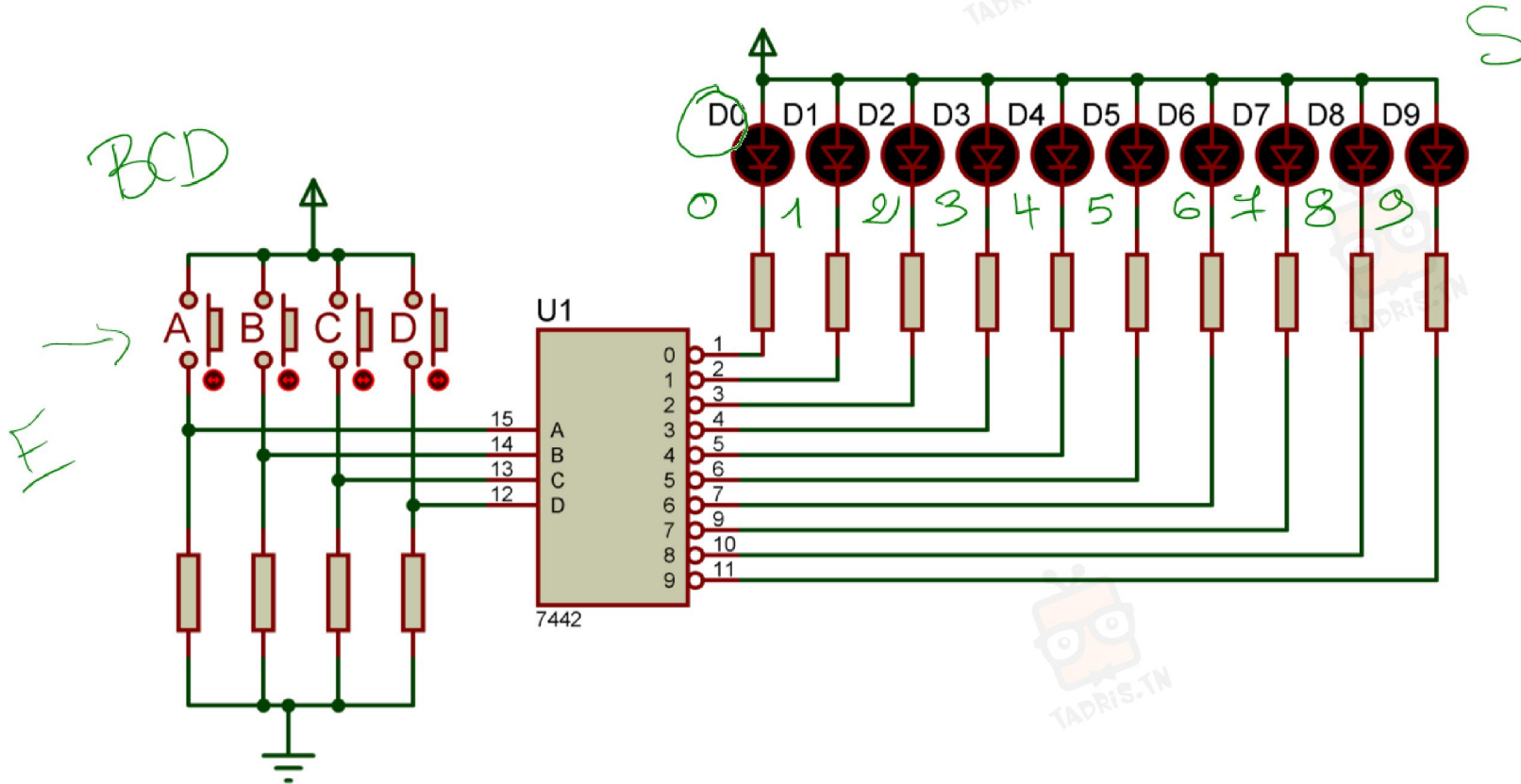
{
    if (portb.b7) S2 = 9;
    else if (portb.b6) S2 = 8;
    else if (portb.b5) S2 = 7;
    else if (portb.b4) S2 = 6;
    else if (portb.b3) S2 = 5;
    else if (portb.b2) S2 = 4;
    else if (portb.b1) S2 = 3;
    else if (portb.b0) S2 = 2;
    else if (portA.b4) S2 = 1;
    else S2 = 0;
    portA = S2; affichage sur le portA
}
}

```



Exercice 2: Fonction décodage BCD/Décimal

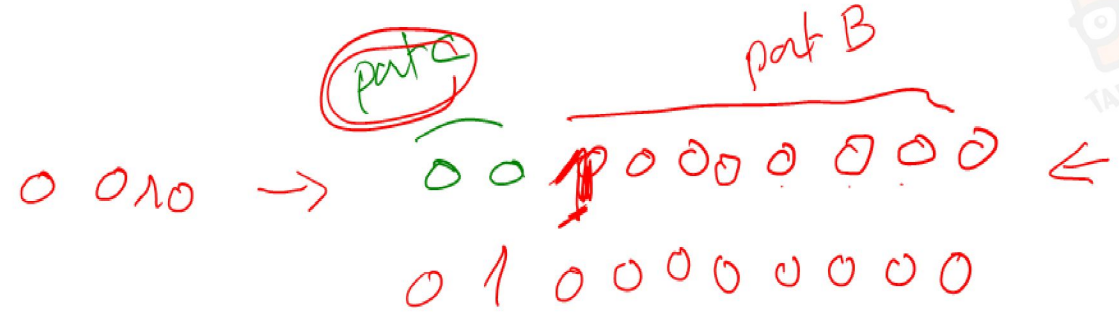
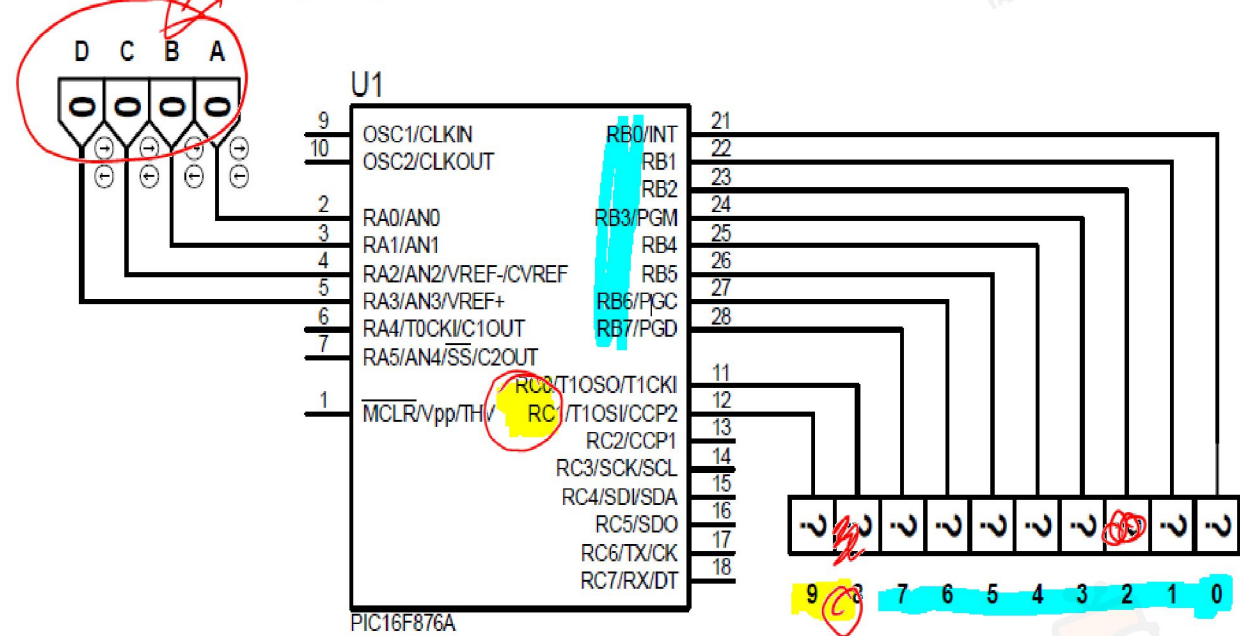
1- Compléter le tableau en se référant au montage du circuit 7442.



ENTREES				SORTIES
D	C	B	A	
0	0	0	0	D ₀
0	0	0	1	D ₁
0	0	1	0	D ₂
0	0	1	1	D ₃
0	1	0	0	D ₄
0	1	0	1	D ₅
0	1	1	0	D ₆
0	1	1	1	D ₇
1	0	0	0	D ₈
1	0	0	1	D ₉

7442

Solution programmée avec la structure switch



Compléter le programme en mikroC

```

void main()
{
    TRISA = 0xFF;
    TRISB = 0x00;
    TRISC = 0x00;
    ADCON1 = 6;
    portB = 0;
    portC = 0;
    while (1)
    {
        switch (portA)
        {

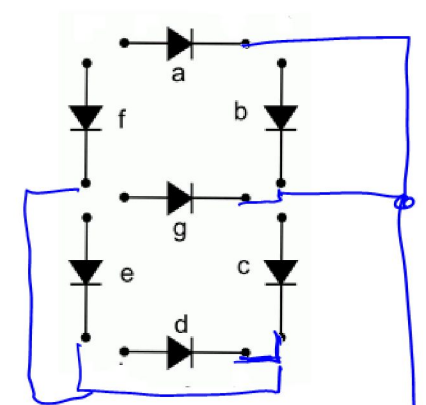
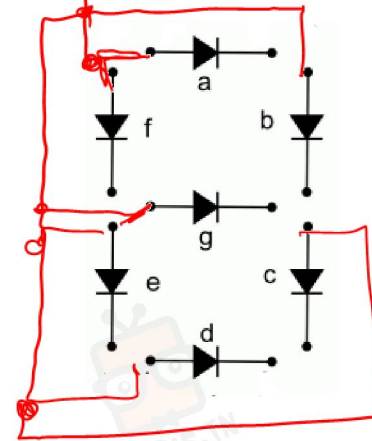
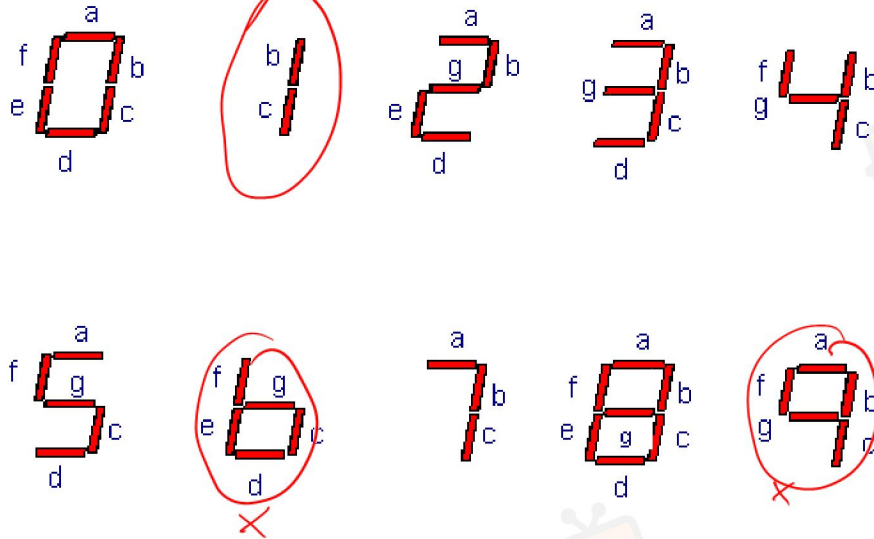
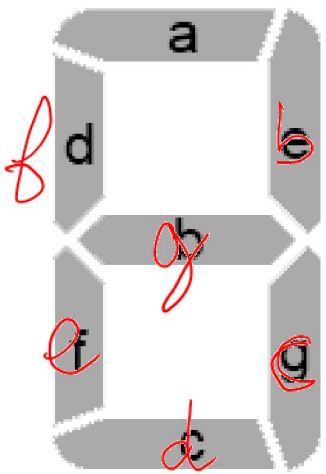
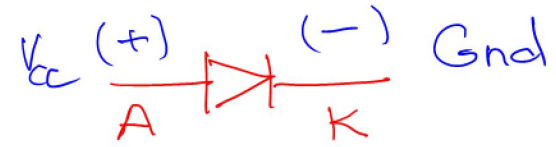
```

```

        case (0x00): portB = 1; portC = 0; break;
        case (0x01): portB = 2; portC = 0; break;
        case (0x02): portB = 4; portC = 0; break;
        case (0x03): portB = 8; portC = 0; break;
        case (0x04): portB = 16; portC = 0; break;
        case (0x05): portB = 32; portC = 0; break;
        case (0x06): portB = 64; portC = 0; break;
        case (0x07): portB = 128; portC = 0; break;
        case (0x08): portB = 0; portC = 1; break;
        case (0x09): portB = 0; portC = 2; break;
        default: portB = 0; portC = 0; break;
    }
}

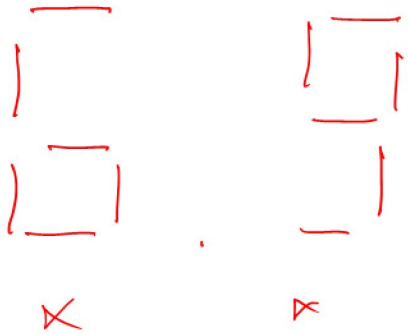
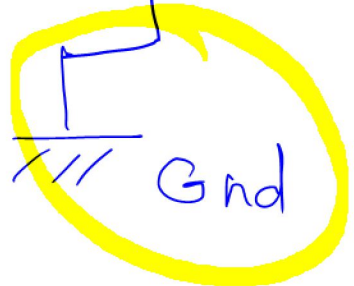
```

Afficheur à sept segments



afficheur à sept segments à **anode** commune

afficheur à sept segments à **cathode** commune.



Exercice 3:

Identifier ce circuit :

afficheur à sept segments à anode commune

Tension commune :

+Vcc

Les segments sont actifs au niveau

bas "0"

Ce circuit est commandé par :

7447

Identifier ce circuit :

afficheur à sept segments à cathode commune

Tension commune :

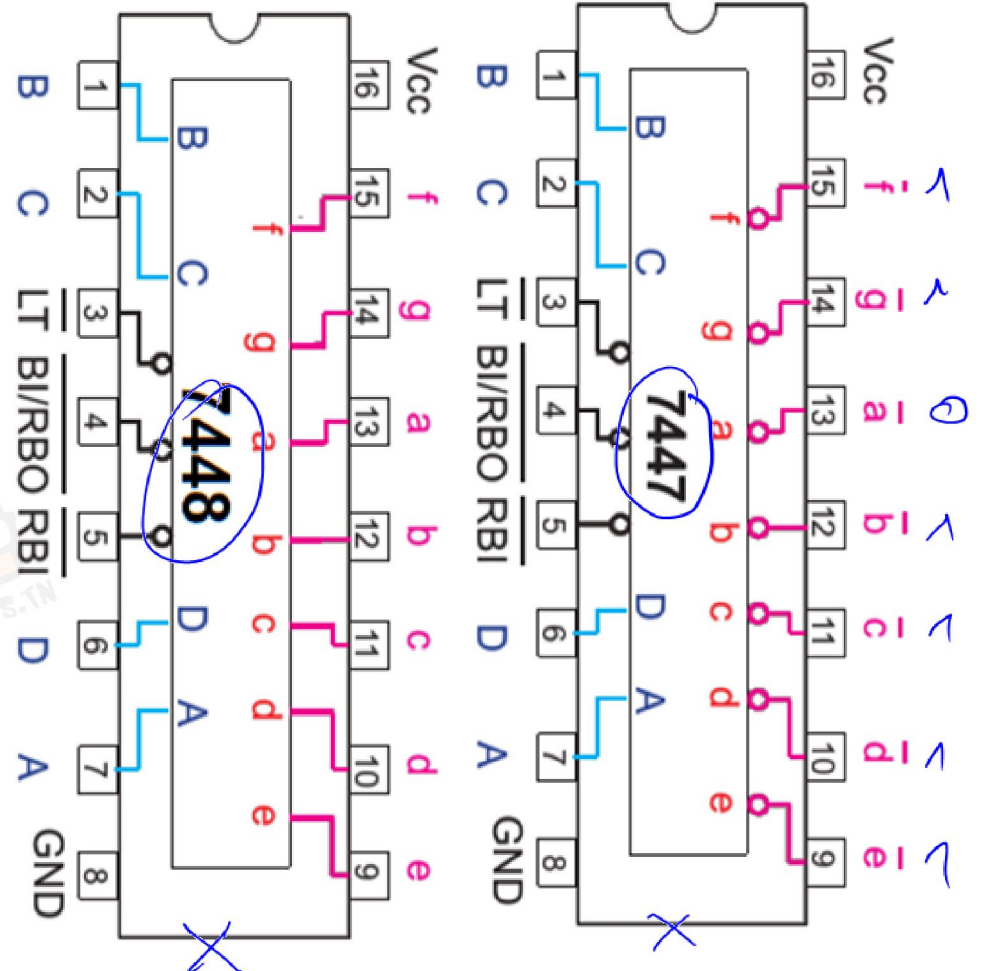
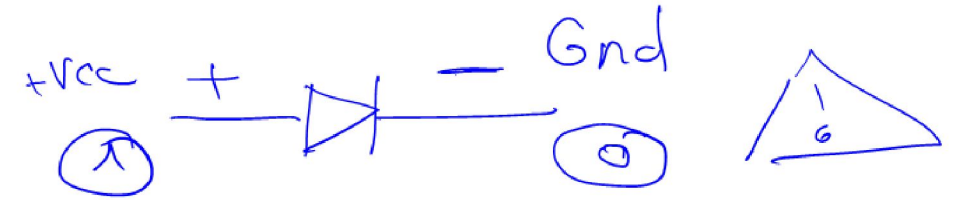
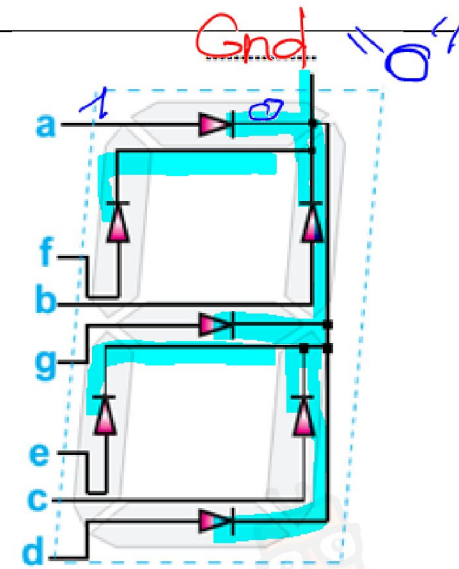
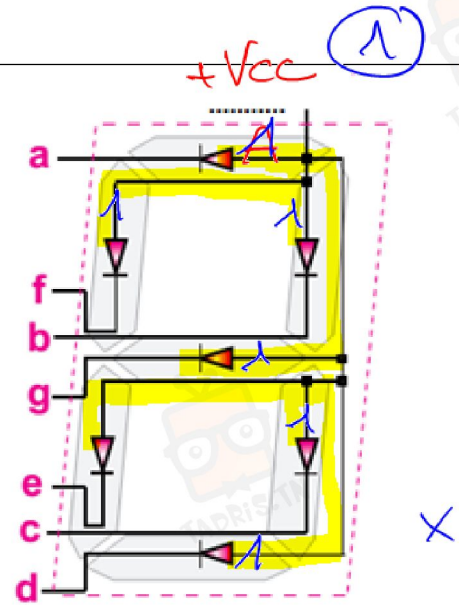
Gnd

Les segments sont actifs au niveau

haut "1"

Ce circuit est commandé par :

7448



Exercice 4:

10 11 12 13 14 15
A B C D E F

Fonction transcodage BCD/7 segments

L'affichage d'un résultat est matérialisé par des afficheurs 7 segments de façon à visualiser les 10 digits comme suit;

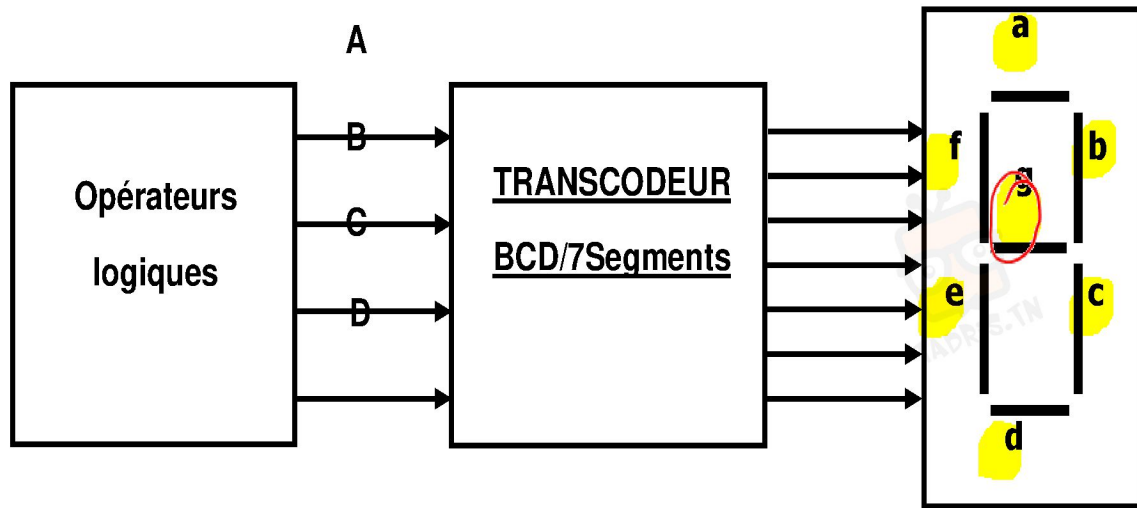
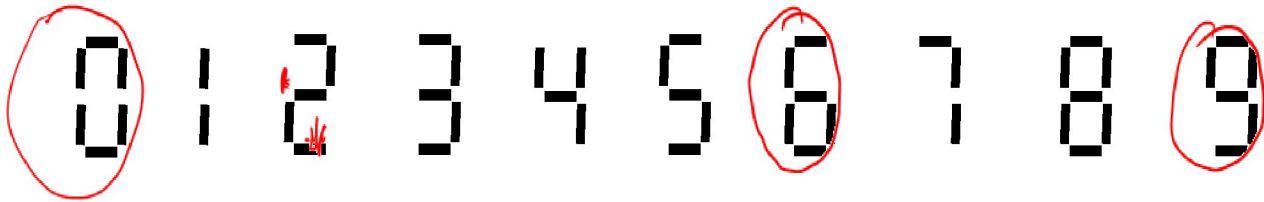


Table de vérité :

Nombres décimaux	Entrées				Sorties						
	D	C	B	A	g	f	e	d	c	b	a
0	0	0	0	0	0	1	1	1	1	1	1
1	0	0	0	1	0	0	0	0	1	1	0
2	0	0	1	0	1	0	1	1	0	1	1
3	0	0	1	1	1	0	0	1	1	1	1
4	0	1	0	0	1	1	0	0	1	1	0
5	0	1	0	1	1	1	0	1	1	0	1
6	0	1	1	0	1	1	1	1	1	0	1
7	0	1	1	1	0	0	0	0	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	0	1	1	1	1

MSB poids plus fort

LSB poids

3F

06

5B

4F

66

6D

7D

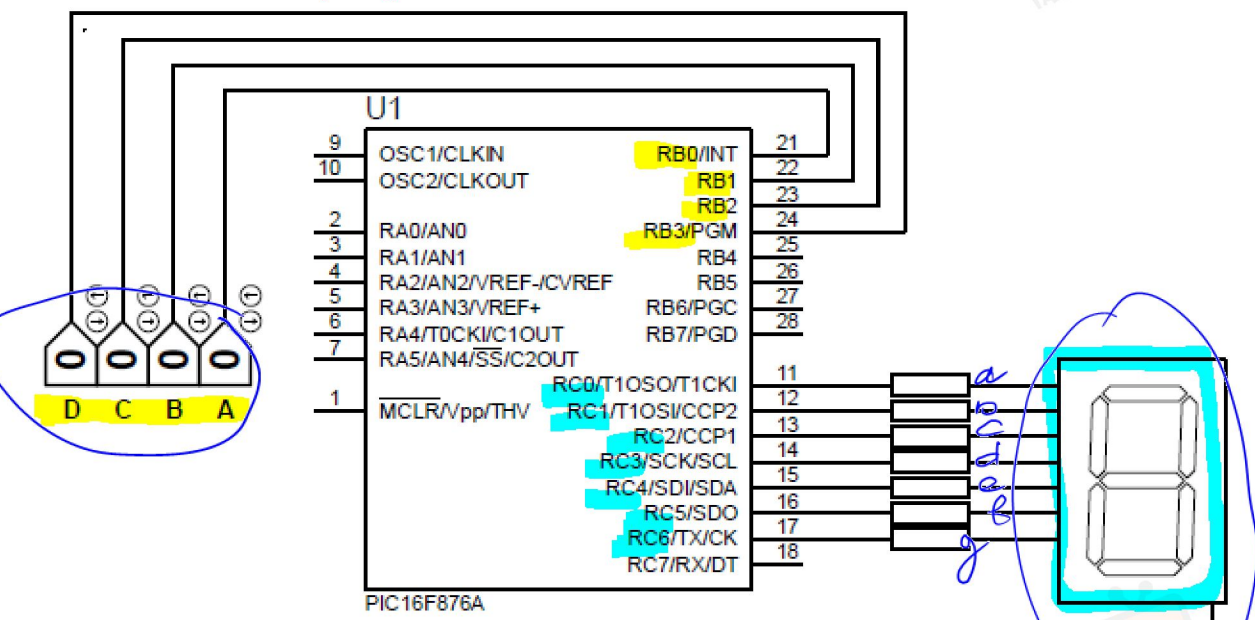
7F

7F

Tris configurations

char 8 bits

Solution programmée



afficheur à 7 segments
à cathode commune
0 → 9

Programme

```

char i at portb ; // Variable d'entrée i reliée au port b
const afficheur[10] = // Tableau de 10 cases du type constantes:
{
0x3F; // valeur de zéro i=0
0x06; // i=1
0x5B;
0x4F;
0x66;
0x6D;
0x7D;
0x07;
0x7F;
0x6F; // i=9
}

```

```

void main() // mot clé programme
{ // début
TrisB = 0xFF; // configuration
TrisC = 0x00;
portc = 0; // initialisation
i = 0; // initialisation
while(1) // boucle infinie
{
portc = afficheur[i]; // portc ← contenu du tableau
}
}

```