

-Programme Arduino avec 7 segments

```
|  
int led2 = 2;  
int led3 = 3;  
int led4 = 4;  
int led5 = 5;  
int led6 = 8;  
int led7 = 9;  
int led8 = 11;  
int Laser = 6;  
int Detector = 7;  
int reset = 10;  
int unit = 0;  
int etat = 0;  
  
void setup()  
{  
  Serial.begin (9600);  
  pinMode(Laser, OUTPUT);  
  pinMode(Detector, INPUT);  
  digitalWrite(reset, HIGH);  
}  
  
int raz()  
{if(digitalRead(reset) == LOW) // Si le bouton reset est appuyé alors on remet à zéro  
{unit = 0;  
return 1;}  
}
```

```

void aff() {
switch (unit) {
  case 0 :
  {
digitalWrite(led2, LOW); // Fixe la led2 comme allumée
digitalWrite(led3, LOW); // Fixe la led3 comme allumée
digitalWrite(led4, LOW); // Fixe la led4 comme allumée
digitalWrite(led5, LOW); // Fixe la led5 comme allumée
digitalWrite(led6, LOW); // Fixe la led6 comme allumée
digitalWrite(led7, LOW); // Fixe la led7 comme allumée
digitalWrite(led8, HIGH); // Fixe la led8 comme éteinte
  }
break;
  case 1 :
  {
digitalWrite(led2, HIGH); // Fixe la led2 comme éteinte
digitalWrite(led3, LOW); // Fixe la led3 comme allumée
digitalWrite(led4, LOW); // Fixe la led4 comme allumée
digitalWrite(led5, HIGH); // Fixe la led5 comme éteinte
digitalWrite(led6, HIGH); // Fixe la led6 comme éteinte
digitalWrite(led7, HIGH); // Fixe la led7 comme éteinte
digitalWrite(led8, HIGH); // Fixe la led8 comme éteinte
  }
break;
  case 2 :
  {
digitalWrite(led2, LOW); // Fixe la led2 comme allumée
digitalWrite(led3, LOW); // Fixe la led3 comme allumée
digitalWrite(led4, HIGH); // Fixe la led4 comme éteinte
digitalWrite(led5, LOW); // Fixe la led5 comme allumée
digitalWrite(led6, LOW); // Fixe la led6 comme allumée
digitalWrite(led7, HIGH); // Fixe la led7 comme éteinte
digitalWrite(led8, LOW); // Fixe la led8 comme allumée
  }
break;
  case 3 :
  {
digitalWrite(led2, LOW); // Fixe la led2 comme allumée
digitalWrite(led3, LOW); // Fixe la led3 comme allumée
digitalWrite(led4, LOW); // Fixe la led4 comme allumée
digitalWrite(led5, LOW); // Fixe la led5 comme allumée
digitalWrite(led6, HIGH); // Fixe la led6 comme éteinte
digitalWrite(led7, HIGH); // Fixe la led7 comme éteinte
digitalWrite(led8, LOW); // Fixe la led8 comme allumée
  }
break;
}
}

```

```
case 4 :
{
digitalWrite(led2, HIGH); // Fixe la led2 comme éteinte
digitalWrite(led3, LOW); // Fixe la led3 comme allumée
digitalWrite(led4, LOW); // Fixe la led4 comme allumée
digitalWrite(led5, HIGH); // Fixe la led5 comme éteinte
digitalWrite(led6, HIGH); // Fixe la led6 comme éteinte
digitalWrite(led7, LOW); // Fixe la led7 comme allumée
digitalWrite(led8, LOW); // Fixe la led8 comme allumée
}
break;
case 5 :
{
digitalWrite(led2, LOW); // Fixe la led2 comme allumée
digitalWrite(led3, HIGH); // Fixe la led3 comme éteinte
digitalWrite(led4, LOW); // Fixe la led4 comme allumée
digitalWrite(led5, LOW); // Fixe la led5 comme allumée
digitalWrite(led6, HIGH); // Fixe la led6 comme éteinte
digitalWrite(led7, LOW); // Fixe la led7 comme allumée
digitalWrite(led8, LOW); // Fixe la led8 comme allumée
}
break;
case 6 :
{
digitalWrite(led2, LOW); // Fixe la led2 comme allumée
digitalWrite(led3, HIGH); // Fixe la led3 comme éteinte
digitalWrite(led4, LOW); // Fixe la led4 comme allumée
digitalWrite(led5, LOW); // Fixe la led5 comme allumée
digitalWrite(led6, LOW); // Fixe la led6 comme allumée
digitalWrite(led7, LOW); // Fixe la led7 comme allumée
digitalWrite(led8, LOW); // Fixe la led8 comme allumée
}
break;
case 7 :
{
digitalWrite(led2, LOW); // Fixe la led2 comme allumée
digitalWrite(led3, LOW); // Fixe la led3 comme allumée
digitalWrite(led4, LOW); // Fixe la led4 comme allumée
digitalWrite(led5, HIGH); // Fixe la led5 comme éteinte
digitalWrite(led6, HIGH); // Fixe la led6 comme éteinte
digitalWrite(led7, HIGH); // Fixe la led7 comme éteinte
digitalWrite(led8, HIGH); // Fixe la led8 comme éteinte
}
break;
```

```

    case 8 :
    {
digitalWrite(led2, LOW); // Fixe la led2 comme allumée
digitalWrite(led3, LOW); // Fixe la led3 comme allumée
digitalWrite(led4, LOW); // Fixe la led4 comme allumée
digitalWrite(led5, LOW); // Fixe la led5 comme allumée
digitalWrite(led6, LOW); // Fixe la led6 comme allumée
digitalWrite(led7, LOW); // Fixe la led7 comme allumée
digitalWrite(led8, LOW); // Fixe la led8 comme allumée
    }
break;
    case 9 :
    {
digitalWrite(led2, LOW); // Fixe la led2 comme allumée
digitalWrite(led3, LOW); // Fixe la led3 comme allumée
digitalWrite(led4, LOW); // Fixe la led4 comme allumée
digitalWrite(led5, LOW); // Fixe la led5 comme allumée
digitalWrite(led6, HIGH); // Fixe la led6 comme éteinte
digitalWrite(led7, LOW); // Fixe la led7 comme allumée
digitalWrite(led8, LOW); // Fixe la led8 comme allumée
    }
break;
default:
raz(); // En cas de problème on force tout à 0
}
}

int compt(){
    if(unit < 9)
    {
boolean Laser = digitalRead(Detector); // mesure du capteur
delay(5); // Temps d'attente de 5ms
boolean etat = Laser; // Retranscription de la mesure pas HIGH (état haut (1) ) ou LOW (état bas(0) )
if (etat == LOW) // Si l'état est bas (bouton reset appuyé)
{unit ++; // On rajoute +1 au nombre de passage unit++ == unit=unit+1
delay(400);} // temps d'attente de 400ms pour ne pas créer de fausses mesures
    }
else
{unit = 0;} // Si il n'y a pas d'état bas alors on ne fait rien
}

void loop()
{
digitalWrite(Laser, HIGH);
do{
aff();
compt();
Serial.println(unit);
delay(1000);
}
while (raz()!=0); // Temps que la remise à zéro n'est pas activée alors le programme de comptage et d'affichage sont lancés
}

```

- Programme Arduino LCD

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,20,4);
int Laser = 6;
int Detector = 7;
int reset = 10;
int unit = 0;
int etat = 0;

void setup()
{
  Serial.begin (9600);
  lcd.init();
  lcd.backlight();
  pinMode(Laser, OUTPUT);
  pinMode(Detector, INPUT);
  digitalWrite(reset, HIGH);
}

int raz()
{if(digitalRead(reset) == LOW) // Si le bouton reset est appuyé alors on remet à zéro
{unit = 0;
  lcd.init();
return 1;}}

}

void aff() {
switch (unit) {
  case 0 :
  {
    lcd.setCursor(0,0);
    lcd.print("Nombre de");
    lcd.setCursor(0,1);
    lcd.print("Personne 0");
  }
  break;
}
```

```
case 1 :
{

lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 1");
}
break;
case 2 :
{

lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 2");
}
break;
case 3 :
{

lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 3");
}
break;
case 4 :
{

lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 4");
}
break;
case 5 :
{

lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 5");
```

```
    }  
    break;  
    case 6 :  
    {  
    lcd.setCursor(0,0);  
    lcd.print("Nombre de");  
    lcd.setCursor(0,1);  
    lcd.print("Personne 6");  
    }  
    break;  
    case 7 :  
    {  
    lcd.setCursor(0,0);  
    lcd.print("Nombre de");  
    lcd.setCursor(0,1);  
    lcd.print("Personne 7");  
    }  
    break;  
    case 8 :  
    {  
    lcd.setCursor(0,0);  
    lcd.print("Nombre de");  
    lcd.setCursor(0,1);  
    lcd.print("Personne 8");  
    }  
    break;  
    case 9 :  
    {  
    lcd.setCursor(0,0);  
    lcd.print("Nombre de");  
    lcd.setCursor(0,1);  
    lcd.print("Personne 9");  
    }  
    break;  
    case 10 :  
    {  
    lcd.setCursor(0,0);  
    lcd.print("Nombre de");  
    lcd.setCursor(0,1);  
    lcd.print("Personne 10");
```

```
}
break;
case 11 :
{
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 11");
}
break;
case 12 :
{
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 12");
}
break;
case 13 :
{
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 13");
}
break;
case 14 :
{
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 14");
}
break;
case 15 :
{
```



```
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 15");
}
break;
case 16 :
{
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 16");
}
break;
case 17 :
{
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 17");
}
break;
case 18 :
{
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 18");
}
break;
case 19 :
{
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
```

```

lcd.print("Personne 19");
}
break;
case 20 :
{
lcd.backlight();
lcd.setCursor(0,0);
lcd.print("Nombre de");
lcd.setCursor(0,1);
lcd.print("Personne 20");
lcd.init();
}
break;
default:
raz(); // En cas de problème on force tout à 0
}
}

int compt(){
if(unit < 20 )
{
boolean Laser = digitalRead(Detector); // mesure du capteur
delay(5); // Temps d'attente de 5ms
boolean etat = Laser; // Retranscription de la mesure pas HIGH (état haut (1) ) ou LOW (état bas(0) )
if (etat == LOW) // Si l'état est bas (bouton reset appuyé)
{unit ++; // On rajoute +1 au nombre de passage unit++ == unit=unit+1
delay(400);} // temps d'attente de 400ms pour ne pas créer de fausse mesures
}
else
{unit = 0;} // Si il n'y a pas d'état bas alors on ne fait rien
}

void loop()
{
digitalWrite(Laser, HIGH);
do{
compt();
aff();

Serial.println(unit);
delay(100);
}
while (raz()!=0); // Temps que la remise à zéro n'est pas activée alors le programme de comptage et d'affichage sont lancés
}

```
