```
1. age = int(input("Quel âge as tu ?"))
2. if age<18:
3.         print("tu es mineur.")
4. else:
5.         print(" tu es majeur.")
```

Explications :

1.définition d'une variable nommée âge
2.si la variable < 18
3.écrire « tu es mineur »
4.sinon:
5.écrire « tu es majeur »

```
1
2 n_I2 = 4     # en mol
3 n_thio = 6     # en mol
4 nbreStoechio_I2 = 1
5 nbreStoechio_thio = 2
6 xmax_I2 = n_I2/nbreStoechio_I2
7 xmax_thio =n_thio/nbreStoechio_thio
8 if (xmax_I2< xmax_thio):
9     print ("le réactif limitant est le diode.")
10    xmax = xmax_I2
11 else :
12    print ("le réactif limitant est l'ion thiosulfate.")
13    xmax = xmax_thio
```

Explications :

1.
2.n_I2 = 4 mol
3.n_thio = 6 mol
4.nombre stoechiométrique I2 = 1
5.nombre stoechiométrique thio = 2
6.calcul de xmax_I2
7.calcul de xmax_thio
8.Si xmax_I2 < xmax_thio
9.écrire «le réactif limitant est le diode »
10.on définit xmax = xmax_I2
11.sinon :
12.écrire « le réactif limitant est l'ion thiosulfate »

```
1.import numpy as np
2. import matplotlib.pyplotasplt
3. x = np.array([0,1,2,3,4,5])
4. y = np.array([0.,1.15,2,3,4.1,5.3])

5.plt.plot(x,y, 'ro', label='points expérimentaux')
6.modele = np.polyfit(x,y,1)

7. a,b = [coef for coef in modele]
8. plt.plot(X,a*X+b, 'b-', label='modelisation')

9. print('y=', round(a,3), 'x+(', round(b,3),') ')
10. plt.grid()
11. plt.legend()
```

Explications :

1.importation de la bibliothèque numpy en np
2.importation de la bibliothèque liée au graphe
3.np.array permet de créer des lignes de tableaux de valeurs dont la variable est X
4.puis Y
5.trace le graphe des points expérimentaux X et Y en rond et rouge
6.
7.
8.permet de tracer un graphique aX+b en trait bleu
9.
10. affiche une grille
11. affiche la légende
12. affiche la droite

```
1. import matplotlib.pyplot as plt
#paramètre de la chronophotographie

2. n=4
3. dt=0.035

4. for k in range (0, n-1) :
5.        vy = (y[k+1] - y[k]) / dt
6.        echelle = 0.02
7.        vy = vy * echelle
8.       plt.quiver (x[k], y[k], 0, vy, color =
```

Explications :

1.Importation de la bibliothèque liée au graphe

2. définition de n=4
3. Définition de dt = 0,035

4. Permet une boucle répète instruction un certain nombre de fois.
5. cette instruction permet de calculer la valeur de vitesse Vy pour chaque k entre 0et n-2
6.
7. défini l'échelle de représentation des vecteurs vitesse
8. permet de tracer un vecteur $\vec{V}$ dont origine est M([illegible]

```
1. def esp (na, nb, S, T, U, V) :
2. x1 = na / S
3. x2 = nb / T
4. R = min (x1, x2)

5. print(« l'état final de A est :" , na- S*R)
6. print(« l'état final de B est :" , nb-T*R)
7. print(« l'état final de C est :", U*R)
8. print(« l'état final de D est :" , V*R)
```

Explications :

1.définition des variables et
2. leur valeur…
3.
4.

5. écrire l'état final de A est: , na -S*R
6. écrire l'état final de B est : , nb -T*R
7. écrire l'état final de C est : , U*R
8. écrire l'état final de D est : , V*R

```
1. import numpy as np
from spicy import *
2. import matplotlib.pyplot as plt
3. T=
4. a=
5. b=
6. c=
7. t=np.linspace(0,T,22)
8. g=9.8
9. x=0-g/2/a*t
10. y=c+b*t+a*t**2
11. plt.plot((x,y, «o»)
12. plt.grid()
```

Explications :

Tracé des points modélisant la trajectoire parabolique étudiée
1.importation de la bibliothèque numpy en np
2.importation de la bibliothèque liée au graphe.
3.T=
4.a=
5.b=
6.c=
7.les 22 points appartiennent à l'intervalle 0 et T
8.g= 9,8
9.x= equation horaire de x
10. y = equation horaire de y
11. affiche les points x,y,en rond
12. affiche une grille sur le graphe

```
pyplot.arrow(
        x[i],y[i], Vx[i]/echelle,
Vy[i]/echelle,
        head_width=0,01,
        length_includes_head=true
)
```

Explications :

```
# trace le vecteur vitesse du point Mi

# coordonnées du point de départ
# longueur du vecteur sur axe x et axe y
# Largeur de la flèche
# la longueur inclut la flèche
```