```
mport matplotlib.pyplot as plt
import numpy as np

#
v_col = 10                          # / ( )
v_mot1 = 20                         # / (  1)
v_mot2 = 15                         # / (  2)
L = 5000                            #  ,

#
t_max = max(L/(v_mot1-v_col), L/(v_mot2+v_col)) * 1.1  #
times = np.linspace(0, t_max, 1000)  #

#
pos_tail = v_col * times             #
pos_head = L + v_col * times         #

#  :
# 1-  (   )
pos_mot1 = v_mot1 * times           #    ,

# 2-  (   )
pos_mot2 = L - v_mot2 * times       #     L

#
intersect_time1 = L / (v_mot1 - v_col)  #   1
intersect_time2 = L / (v_mot2 + v_col)  #   2

#
intersect_time3 = L / (v_mot1 + v_mot2)  #

#
print("\n :")
print("-" * 50)
print(f":")
print(f"1.  1    : {intersect_time1:.2f} .")
print(f"2.  2    : {intersect_time2:.2f} .")
print(f"3.       : {intersect_time3:.2f} .")

#
plt.figure(figsize=(10, 6))          #

#
plt.plot(times, pos_tail, label=' ', color='blue',
↪  linewidth=2)
plt.plot(times, pos_head, label=' ', color='green',
↪  linewidth=2)
plt.plot(times, pos_mot1, label=' 1', color='orange',
↪  linewidth=2)
```

```
    plt.plot(times, pos_mot2, label=' 2', color='purple', linewidth=2)

    #
    plt.vlines(intersect_time1, ymin=-100, ymax=np.interp(intersect_time1, times, pos_head), colors='black', linestyles='dashed')
    plt.text(intersect_time1, -200, f'{intersect_time1:.2f}', ha='center', va='bottom', fontweight='bold')

    plt.vlines(intersect_time2, ymin=-100, ymax=np.interp(intersect_time2, times, pos_tail), colors='black', linestyles='dashed')
    plt.text(intersect_time2, -200, f'{intersect_time2:.2f}', ha='center', va='bottom', fontweight='bold')

    #   |
    plt.vlines(intersect_time3, ymin=-100, ymax=np.interp(intersect_time3, times, pos_mot1), colors='black', linestyles='dashed')
    plt.text(intersect_time3, -200, f'{intersect_time3:.2f}', ha='center', va='bottom', fontweight='bold')

    #
    plt.xlabel(' ()', fontsize=12)
    plt.ylabel(' ()', fontsize=12)
    plt.title('    ', fontsize=14)
    plt.legend(fontsize=10)
    plt.grid(True)

    #    Y   11000
    plt.ylim(-100, 11000)

    #
    plt.show()
```