**Assignment 1**

————Cao Jinhao 122010910216

**Question 1**

Solution:

****  
Maximized profit = 2900 dollars

Product 1 = 26 units

Product 2 = 55 units

Product 3 = 20 units

The Python code part is *Function “Assignment1Question1”*.

**Question 2**

Solution:

1. Create ten instances of TSP problems with different numbers of cities, ranging from 5 to 14 with 1 step ( and two bigger numbers: 20 and 50). *Function “GenerateCoordinate”* is for generating coordinates of cities randomly.
2. Copy mainly the code provided in the class. But we need to change the input C (distance matrix) into coordinate matrix. *Function “Coordinate2Distance”* can do this. In this question, ten distance matrixes are symmetric. You can refer to *Function “Assignment1Question2Method1”.*
3. To finish with all subtour elimination constraints, we need to add extra constraints which are contained in the following formula. We can use *Function “CombineSubtour”* (need to import a basic *Package “itertools”*) to list all subtours. The key codes are from Line 147 to Line 154. You can refer to *Function “Assignment1Question2Method2”.*



1. The difference between Method1 and Method2 is time complexity. Method 1 is faster. Method 2 can solve TSP with less cities (nodes), because listing all subtours will cause exponential explosion. The following table shows the two methods’ efficiency.

|  |  |  |  |
| --- | --- | --- | --- |
| Number of Cities || Time/(s)||Method | Method 1 | Method 2 | Comment |
| 5 | 0.00 | 0.00 | So fast |
| 6 | 0.00 | 0.01 |  |
| 7 | 0.00 | 0.01 |  |
| 8 | 0.00 | 0.01 |  |
| 9 | 0.01 | 0.03 |  |
| 10 | 0.01 | 0.06 | Method 2 processes more slowly. |
| 11 | 0.01 | 0.13 |  |
| 12 | 0.01 | 0.30 |  |
| 13 | 0.01 | 0.71 |  |
| 14 | 0.01 | 1.78 |  |
| 20 | 0.02 | --- | Method 2 is hard to exit the debug process. |
| 50 | 0.07 | --- | Method 2 can not get the result. |