Conveyor Belt

In a special, circular factory building, a conveyor belt has been put into operation for the transportation of various materials along a circular path. Any material can be placed on this conveyor belt in a compartment, with the setting of its destination. At the destination, the compartment is automatically removed from the belt. The belt moves in the direction of clockwise rotation, starting when a compartment is placed on it, and it continues moving until there are compartments to be transported on it. These are the tasks it needs to handle.

You have the data file named "szallit.txt" available, where the first line contains two integers separated by a space. The first integer represents the length of the conveyor belt, and the second one represents the time required for the conveyor belt to move one unit. In the following, up to 1000 lines, each line contains four integers separated by spaces. The first number indicates when a compartment was placed on the belt. The next two numbers show the origin and destination of the transportation (the distance traveled is always less than the length of the belt). The last number represents the weight of the transported item. Length and positions are given in distance units, time is given in time units, and weight is given in weight units. None of the data in the data file exceeds 500.

```
200 3
1 134 64 34
14 22 129 83
14 135 54 21
23 31 40 61
```

In the example, the first 5 lines of the file are shown. The first line indicates that the conveyor belt is 200 units long, and it travels one unit of distance in 3 time units. The 2nd line contains the data for the first compartment. According to the 3rd line, at the 14th time unit, a compartment is placed at the position 22 distance units, and it is removed at the position 129 distance units. The compartment weighs 83 units. Therefore, the compartment travels a distance of 129-22=107 units, and it arrives at its destination at time 14+107*3=335.

The next line describes another compartment that is also placed on the belt at the 14th time unit, traveling from position 135 to position 54. The distance traveled is, therefore, 200-135 units initially, and then 54 units, for a total distance of 119 units, and the weight of this compartment is 21 units.

Create a program that uses the data from the file to answer the following questions! Save the source code of the program as "belt". When writing the program, you do not need to check the correctness or validity of the data provided by the user, and you can assume that the available data conform to the description.

1. Read the data from the "szallit.txt" file and solve the following tasks using it!
2. Ask for the number of a shipment, and then display its departure and destination locations! (Shipments are numbered from 1.)
3. Create a function named "tav" that calculates the distance of the shipment on the conveyor belt based on the length of the belt, as well as the knowledge of the departure and destination locations. Use this function in solving subsequent tasks. Create the function header according to the following specifications and use the variable names provided in the solution.

Method tav(szalaghossz, indulashelye, erkezeshelye : int): int

This function should take the conveyor belt's length (`salag_hossz`), departure location (`indulasi_hely`), and destination location (`celhely`) as parameters and return the distance traveled by the shipment.

4. Determine the maximum transportation distance during the available shipments! Print on the screen the maximum distance and the numbers of all shipments that had this maximum distance!
5. Specify how much weight in total passed in front of position 0! Do not consider compartments that either originated from or arrived at that position!
6. Prompt for a timestamp and determine the numbers of the compartments that were transported at that specific timestamp! Include the compartments that were just starting their journey at that moment, but exclude those that have reached their destination by that time. If there were no transported compartments at that timestamp, then display the word "Empty" instead of compartment numbers!
7. Create the "weight.txt" file that specifies the total weight transported from each location! Exclude the locations where no transportation occurred from appearing in the file! (The order of data written in the file can be arbitrary.)