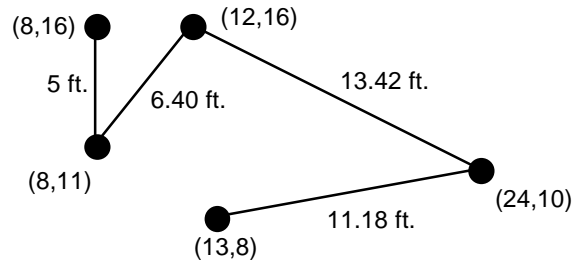


# Problem A

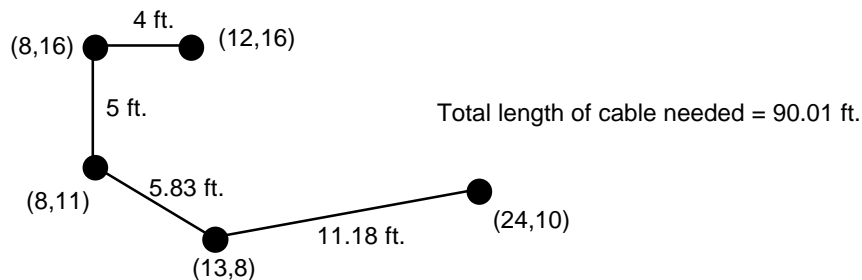
## Getting in Line

Computer networking requires that the computers in the network be linked. This problem considers a “linear” network in which the computers are chained together so that each is connected to exactly two others except for the two computers on the ends of the chain which are connected to only one other computer. A picture is shown below. Here the computers are the black dots and their locations in the network are identified by planar coordinates (relative to a coordinate system not shown in the picture). Distances between linked computers in the network are shown in feet.



For various reasons it is desirable to minimize the length of cable used. Your problem is to determine how the computers *should be* connected into such a chain to minimize the total amount of cable needed. In the installation being constructed, the cabling will run beneath the floor, so the amount of cable used to join 2 adjacent computers on the network will be equal to the distance between the computers plus 16 additional feet of cable to connect from the floor to the computers and provide some slack for ease of installation.

The picture below shows the optimal way of connecting the computers shown above, and the total length of cable required for this configuration is  $(4+16) + (5+16) + (5.83+16) + (11.18+16) = 90.01$  feet.



### Input

The input file will consist of a series of data sets. Each data set will begin with a line consisting of a single number indicating the number of computers in a network. Each network has at least 2 and at most 8 computers. A value of 0 for the number of computers indicates the end of input. After the initial line in a data set specifying the number of computers in a network, each additional line in the data set will give the coordinates of a computer in the network. These coordinates will be integers in the range 0 to 150. No two computers are at identical locations and each computer will be listed once.

### Output

The output for each network should include a line which tells the number of the network (as determined by its position in the input data), and one line for each length of cable to be cut to connect each adjacent pair of computers in the network. The final line should be a sentence indicating the total amount of cable used. **In listing the lengths of cable to be cut, traverse the network from one end to the other.** (It makes no difference at which end you start.) Use a format similar to the one shown in the sample output, with a line of asterisks separating output for different networks and with distances in feet printed to 2 decimal places.

UW-Madison 2003-2004 ACM-ICPC Placement Test (page 2 of 10)  
1992 World Finals Problem B (page 2 of 2)

Sample Input

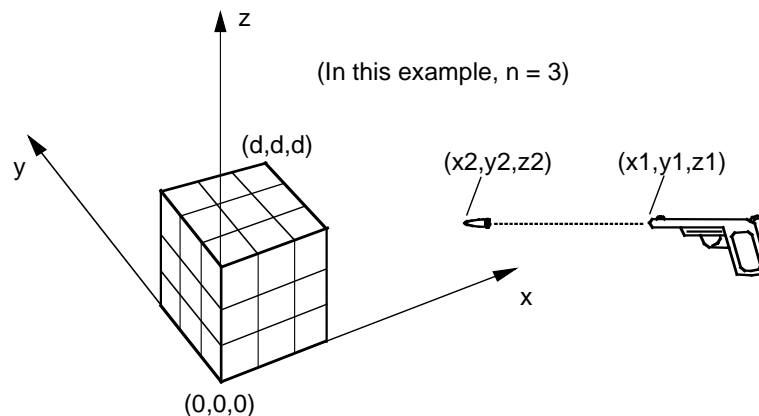
```
6
5 19
55 28
38 101
28 62
111 84
43 116
5
11 27
84 99
142 81
88 30
95 38
3
132 73
49 86
72 111
0
```

Output for the Sample Input

```
*****
Network #1
Cable requirement to connect (5,19) to (55,28) is 66.80 feet.
Cable requirement to connect (55,28) to (28,62) is 59.42 feet.
Cable requirement to connect (28,62) to (38,101) is 56.26 feet.
Cable requirement to connect (38,101) to (43,116) is 31.81 feet.
Cable requirement to connect (43,116) to (111,84) is 91.15 feet.
Number of feet of cable required is 305.45.
*****
Network #2
Cable requirement to connect (11,27) to (88,30) is 93.06 feet.
Cable requirement to connect (88,30) to (95,38) is 26.63 feet.
Cable requirement to connect (95,38) to (84,99) is 77.98 feet.
Cable requirement to connect (84,99) to (142,81) is 76.73 feet.
Number of feet of cable required is 274.40.
*****
Network #3
Cable requirement to connect (132,73) to (72,111) is 87.02 feet.
Cable requirement to connect (72,111) to (49,86) is 49.97 feet.
Number of feet of cable required is 136.99.
```

## Problem B

### Bullet Hole



A cube is suspended in space. A Cartesian coordinate system is defined with its origin at one of the bottom corners of the cube, as shown in the figure. The cube has side dimension  $d$ , so its opposite corners are at coordinates  $(0, 0, 0)$  and  $(d, d, d)$ . The positive  $z$ -direction of the coordinate system is “up” with respect to gravity.

The interior of the cube contains partitions with uniform spacing in each dimension, so that the cube is partitioned into  $n^3$  mini-cubes of equal size. The partitions are thin and watertight, and each mini-cube is filled with water. The total volume of water in all the minicubes is  $d^3$ .

A gun fires a bullet which may hit the cube. The muzzle of the gun is at the point  $(x_1, y_1, z_1)$ . The point  $(x_2, y_2, z_2)$  is a point on the bullet’s path that defines the direction of the bullet. The bullet does not shatter the cube, but wherever the bullet touches a side or interior partition of the cube, it makes a small hole. Bullet holes may be made in the sides, edges, or corners of the interior mini-cubes. Water, influenced by gravity, may leak through these small holes. All the water that leaks out of the large cube is collected and measured.

#### Input

The input data set consists of several trials. Each trial is described by eight integers. The first integer is  $n$  ( $n \leq 50$ ), as described above. The second integer is  $d$  ( $d \leq 100$ ). The remaining six integers— $x_1, y_1, z_1, x_2, y_2, z_2$ —represent the origin and a point on the path of the bullet ( $-100 \leq x_1, y_1, z_1, x_2, y_2, z_2 \leq 100$ ). The origin and the point on the path of the bullet are not the same. After the last trial, the integer 0 terminates the data set.

#### Output

Your program must compute the total volume of water that leaks out of the large cube. For each trial, print the trial number, the notation `Volume =`, and the total volume of water accurate to two digits to the right of the decimal point. Print a blank line between trials.

#### Note

In this problem, two real numbers are considered equal if they are less than  $10^{-6}$  apart.

**Sample Input**

```
5 25 5 15 0 5 15 100
3 30 0 -35 0 3 -25 3
10 16 8 17 11 12 19 6
0
```

**Output for the Sample Input**

```
Trial 1, Volume = 2500.00

Trial 2, Volume = 1950.00

Trial 3, Volume = 0.00
```

# Problem C

## Queue and A

The customer support group of Contest.com receives and responds to requests for technical support via e-mail. Requests may begin arriving when the office opens at 8:00 a.m. and all requests must be serviced by the end of the day.

As requests are received, they are classified according to a predetermined list of topics. Each member of the support staff has responsibility for one or more of these topics and each topic has one or more support personnel assigned to it. Because staff members have different levels of expertise, each staff member has a prioritized list of topics that he or she can handle. Staff personnel are not permitted to handle requests outside their specified areas.

As staff members become available, they select from the pool of waiting requests according to their priority list of topics. All requests arriving at time  $t$  are available for allocation at time  $t$ . If two staff members are simultaneously available, scheduling preference is given to the one whose most recent job was scheduled earliest. If there is still a tie, scheduling preference is given to the person whose id number appears earlier in the input list of staff people. At the opening of business, all personnel are available to handle requests.

You have been asked to perform a preliminary analysis of the technical support environment based on a number of different scenarios. For each scenario, information will be given about the mix of requests and the division of labor among the staff. For each topic, you will be given the average number of requests per day for that topic, the average elapsed time before the first of these requests is received, the average time between requests for this topic, and the average time needed to service the request. All times are given in minutes. You will also be given a list of support personnel and, for each one, a list of the topics for which he or she has responsibility. (Since data are based on estimates, factors such as coffee breaks, lunch, computer failures, etc., can be ignored.)

### Input

Input consists of a number of scenarios. Each scenario begins with the number of request topics, a positive integer no larger than 20. This is followed by a description of each topic. Each description consists of five integer values: a unique topic identifier, the number of requests for that topic, the elapsed time before the first request for that topic is received, the time needed to service a request, and the time between successive requests. All but the third of these values are positive integers; the elapsed time until the first request could be zero. Following this, the number of personnel is given. This will be a positive integer not to exceed 5. Finally, a description of each person is given in the form of three or more positive integer values: a unique identifying number for the person, the number of topics covered by this person, and a list of the topic identifiers arranged from highest priority to lowest priority for that person. A zero follows the last scenario.

### Output

For each scenario, the output consists of the scenario number followed by the statement, "All requests are serviced within  $m$  minutes," where  $m$  is the number of minutes from the start of the business day until the last request is serviced.

**Sample Input**

**Output for the Sample Input**

<pre>3 128 20 0 5 10 134 25 5 6 7 153 30 10 4 5 4 10 2 128 134 11 1 134 12 2 128 153 13 1 153 0</pre>	<pre>Scenario 1: All requests are serviced within 195 minutes.</pre>
---	--

## Problem D

### Flight Planning

Your job is to write a program that plans airplane flights. Each flight consists of a series of one or more legs. Your program must choose the best altitude for each leg to minimize the amount of fuel consumption during the trip.

The airplane has a fixed airspeed, given by the constant `VCRUISE`, and a most-efficient cruising altitude, `AOPT`. When flying at altitude `AOPT`, fuel consumption in gallons per hour is given by `GPHOPT`. When flying at an altitude that is different from `AOPT`, fuel consumption increases by `GPHEXTRA` for each 1000 feet above or below `AOPT`. The flight starts and finishes at an altitude of 0. Each 1000 foot climb burns an extra amount of fuel given by `CLIMBCOST` (there is no reduction in fuel consumption when you descend). Make the approximation that all climbing and descending is done in zero time at the beginning of each flight leg. Thus each leg is flown at a constant airspeed and altitude. For this problem, the airplane characteristics are given by the following constants:

<code>VCRUISE</code>	400 knots (a knot is one nautical mile per hour)
<code>AOPT</code>	30,000 feet
<code>GPHOPT</code>	2000 gallons per hour
<code>GPHEXTRA</code>	10 gallons per hour for each 1000 feet
<code>CLIMBCOST</code>	50 gallons per 1000 feet of climb

Before each flight, you are given the length of each leg and the tailwind expected for each leg. A positive tailwind increases the airplane's speed over the ground, and a negative tailwind decreases its speed over the ground. For example, if airspeed is 400 knots and the tailwind is -50 knots, speed over the ground is 350 knots.

By policy, altitude for each leg must be some integer multiple of 1000 feet, between 20,000 and 40,000 feet, inclusive. Your program must compute the best altitude for each leg to minimize overall fuel consumption for the trip, and must compute the fuel required for the trip.

#### Input

The first line contains an integer `N`, representing the number of flights you are required to plan. Each flight consists of the following input lines:

The first input line in a flight contains an integer `K` ( $0 < K < 10$ ), representing the number of legs in the flight.

The next `K` input lines each contain the following three integers:

- (1) The length of the leg, in nautical miles
- (2) The expected tailwind at 20,000 feet, in knots
- (3) The expected tailwind at 40,000 feet, in knots

The expected tailwind at altitudes between 20,000 and 40,000 feet is computed by linear interpolation. For example, the expected tailwind at 30,000 feet is halfway between the expected tailwind at 20,000 feet and the expected tailwind at 40,000 feet.

#### Output

Your program must produce one output line for each flight. The output line must contain the flight number (counting from the beginning of the problem), the chosen altitude for each leg (in thousands of feet), and the fuel required for the trip (in gallons, to the nearest gallon).

**Sample Input**

```
2
2
1500 -50 50
1000 0 0
3
1000 50 0
2000 0 20
1800 -50 100
```

**Output for the Sample Input**

```
Flight 1: 35 30 13985
Flight 2: 20 30 40 23983
```



# Problem E

## Budget Travel

An American travel agency is sometimes asked to estimate the minimum cost of traveling from one city to another by automobile. The travel agency maintains lists of many of the gasoline stations along the popular routes. The list contains the location and the current price per gallon of gasoline for each station on the list.

In order to simplify the process of estimating this cost, the agency uses the following rules of thumb about the behavior of automobile drivers.

- A driver never stops at a gasoline station when the gasoline tank contains more than half of its capacity unless the car cannot get to the following station (if there is one) or the destination with the amount of gasoline in the tank.
- A driver always fills the gasoline tank completely at every gasoline station stop.
- When stopped at a gasoline station, a driver will spend \$2.00 on snacks and goodies for the trip.
- A driver needs no more gasoline than necessary to reach a gasoline station or the city limits of the destination. There is no need for a “safety margin.”
- A driver always begins with a full tank of gasoline.
- The amount paid at each stop is rounded to the nearest cent (where 100 cents make a dollar).

You must write a program that estimates the minimum amount of money that a driver will pay for gasoline and snacks to make the trip.

### Input

Program input will consist of several data sets corresponding to different trips. Each data set consists of several lines of information. The first 2 lines give information about the origin and destination. The remaining lines of the data set represent the gasoline stations along the route, with one line per gasoline station. The following shows the exact format and meaning of the input data for a single data set.

Line 1: One real number — the distance from the origin to the destination

Line 2: Three real numbers followed by an integer

- The first real number is the gallon capacity of the automobile’s fuel tank.
- The second is the miles per gallon that the automobile can travel.
- The third is the cost in dollars of filling the automobile’s tank in the origination city.
- The integer (less than 51) is the number of gasoline stations along the route.

Each remaining line: Two real numbers

- The first is the distance in miles from the origination city to the gasoline station.
- The second is the price (in cents) per gallon of gasoline sold at that station.

All data for a single data set are positive. Gasoline stations along a route are arranged in nondescending order of distance from the origin. No gasoline station along the route is further from the origin than the distance from the origin to the destination. There are always enough stations appropriately placed along the each route for any car to be able to get from the origin to the destination.

The end of data is indicated by a line containing a single negative number.

### Output

For each input data set, your program must print the data set number and a message indicating the minimum total cost of the gasoline and snacks rounded to the nearest cent. That total cost must include the initial cost of filling the tank at the origin. Sample input data for 2 separate data sets and the corresponding correct output follows.

**Sample Input**

```
475.6
11.9 27.4 14.98 6
102.0 99.9
220.0 132.9
256.3 147.9
275.0 102.9
277.6 112.9
381.8 100.9
516.3
15.7 22.1 20.87 3
125.4 125.9
297.9 112.9
345.2 99.9
-1
```

**Output for the Sample Input**

```
Data Set #1
  minimum cost = $27.31
Data Set #2
  minimum cost = $38.09
```