Simulation

Yonghui Wu
Fudan Univ. & Stony Brook SUNY
yhwu@fudan.edu.cn

- In the real world there are many problems that we can simulate the process to solve. Such problems are called simulation problems. For these problems, solution procedures and rules are showed in problem descriptions. Programs must simulate procedures or implement rules based on descriptions.

- Simulation of Direct Statement
- Simulation by Sieve Method
- Simulation by Construction

Simulation of Direct Statement

- Programmers must read such problems carefully, and simulate processes based on descriptions.
- A problem for simulation of direct statement gets harder as the number of rules increases.

Speed Limit

- **Source: ACM Mid-Central USA 2004**
- **IDs for Online Judge: POJ 2017, ZOJ 2176, UVA 3059**

- Bill and Ted are taking a road trip. But the odometer in their car is broken, so they don't know how many miles they have driven. Fortunately, Bill has a working stopwatch, so they can record their speed and the total time they have driven. Unfortunately, their record keeping strategy is a little odd, so they need help computing the total distance driven. You are to write a program to do this computation.

- For example, if their log shows

Speed in miles perhour	Total elapsed time in hours
20	2
30	6
10	7

- this means they drove 2 hours at 20 miles per hour, then 6-2=4 hours at 30 miles per hour, then 7-6=1 hour at 10 miles per hour. The distance driven is then (2)(20) + (4)(30) + (1)(10) = 40 + 120 + 10 = 170 miles. Note that the total elapsed time is always since the beginning of the trip, not since the previous entry in their log.

- **Input**
- The input consists of one or more data sets. Each set starts with a line containing an integer *n*, 1 <= *n* <= 10, followed by *n* pairs of values, one pair per line. The first value in a pair, *s*, is the speed in miles per hour and the second value, *t*, is the total elapsed time. Both *s* and *t* are integers, 1 <= *s* <= 90 and 1 <= *t* <= 12. The values for *t* are always in strictly increasing order. A value of -1 for *n* signals the end of the input.

- **Output**
- For each input set, print the distance driven, followed by a space, followed by the word "miles"

- Analysis
- The problem is a simple problem of direct statement. We can simulate the stopwatch's running to compute the total distance driven: If the last "total elapsed time in hours" is *z*, the current "speed in miles per hour" is *x*, and the current "total elapsed time in hours" is *y*, then the current distance driven is (*y*-*z*)**x*, and add it to the total distance driven.