



Old Mutual - CSSA Computer Olympiad Training DAY 1



Overview

Problem	Score	Jill's Genome	Latin Squares	IOIWari board game
Submitted By	<i>Jacques</i>	<i>Kurt</i>	<i>Hienrich</i>	<i>Graham</i>
Program name	score.exe	gene.exe	latin.exe	ioiwari.exe
Source name	score.pas	gene.pas	latin.pas	ioiwari.pas
	score.cpp	gene.cpp	latin.cpp	ioiwari.cpp
Input file	score.in	gene.in	latin.in	ioiwari.in
Output file	score.out	gene.out	latin.out	ioiwari.out
Time limit per test	5 seconds	1 seconds	5 seconds	5 seconds
Number of tests	10	10	5	25
Points per test	10	10	20	4
Total points	100	100	100	100

The maximum total score for Day1 is 300 points.



Old Mutual - CSSA Computer Olympiad Training Day 1



Score

Author: Jacques

Description

The more points a student score at training camps, the happier we as the training staff will be. We try to design our contests so that people can score as many points as possible, and would like your assistance.

We have several categories from which problems can be chosen, where a "category" is an unlimited set of contest problems which all require the same amount of time to solve and deserve the same number of points for a correct solution.

Task

Your task is write a program which tells our staff how many problems from each category to include in a contest so as to maximize the total number of points in the chosen problems while keeping the total solution time within the length of the contest.

Your program should determine the number of problems we should take from each category to make the highest-scoring contest solvable within the length of the contest. Remember, the number from any category can be any non-negative integer (0, one, or many). Calculate the maximum number of possible points.

Input (*score.in*)

The input includes the length of the contest M , $1 \leq M \leq 10,000$ (don't worry, you won't be forced to take part in the longer contests) and N , the number of problem categories, where $1 \leq N \leq 10,000$.

Each of the subsequent N lines contains two integers describing a category:

the first integer tells the number of points a problem from that category is worth $1 \leq \text{points} \leq 10000$;

the second tells the number of minutes a problem from that category takes to solve $1 \leq \text{minutes} \leq 10000$.

Line 1:

M N , contest length in minutes (M) and number of problem classes (N)

Subsequent lines 2 ... $N+1$:

Two integers, the *points* and *minutes* for each class

Sample input:

score.in

```
300 4
100 60
250 120
120 100
35 20
```

Output (*score.out*)

Sample output:

score.out

```
605
```

Constraints

$1 \leq M \leq 10,000$

$1 \leq N \leq 10,000$

$1 \leq \text{points} \leq 10000$

$1 \leq \text{minutes} \leq 10000$

Time Limit:

Maximum time allowed per test case is 5 seconds

Scoring:

10 points for each correct output.

0 points for each incorrect output.



Old Mutual - CSSA Computer Olympiad Training Day 1



Jill's Genome

Author: Kurt

Description

Jill (of “Jack & Jill” fame) has recently graduated from Stanford University with a degree in biochemistry. Being top of her class, Jill has been snapped up by Chemtex, a large medical research facility, to continue their research of the human genome.

Genes are encoded by a string containing the four characters “ACGT”, corresponding to the four molecules which constitute DNA. When finding new sequences, it is Jill’s task to determine which existing sequence it is most similar to. Two sequences are most similar if they share the longest possible subsequence of molecules (a.k.a. “ACGT”).

A subsequence of a gene is represented by sub-string of characters that form part of the gene, but are not necessarily successive characters, e.g. in the gene “ACTG”, the subsequence “ATG” would be entirely viable. Similarly in the gene “GATCGAT”, the subsequence “GTGT” would be acceptable.

Task

Jill’s task is to determine how similar two genes are by finding the longest possible subsequence common to both genes.

Input (gene.in)

The input file, gene.in, will consist of four lines. The first and third lines will consist of single integer values, M and N respectively, which represents the lengths of the first and second genes. The second and fourth lines consist of a string of characters, which represent the respective genes.

Sample input:

gene.in

```
4
ACGT
8
AGCTGCAT
```

Output (gene.out)

The output file, gene.out, must consist of one line. The first line of output will consist of one integer value, K, which represents the length of the longest common subsequence.

Sample output:

gene.out

```
4
```

Constraints

$1 \leq M \leq 65,536$

$1 \leq N \leq 65,536$

Time Limit:

Maximum time allowed per test case is 5 seconds

Scoring:

10 points for each correct output.
0 points for each incorrect output.



Old Mutual - CSSA Computer Olympiad Training Day 1



Latin Squares

Author: Hienrich

Description

A square arrangement of numbers

1	2	3	4	5
2	1	4	5	3
3	4	5	1	2
4	5	2	3	1
5	3	1	2	4

is a 5 x 5 Latin Square because each whole number from 1 to 5 appears once and only once in each row and column.

Task

Write a program that will compute the number of $N \times N$ Latin Squares whose first row is:

1 2 3 4 5.....N

Your program should work for any N from 2 to 7 inclusive.

Input (latin.in)

One line containing the integer N.

Sample input:

latin.in

5

Output (latin.out)

A single integer giving the number of latin squares whose first row is:

1 2 3 ... N.

Sample output:

latin.out

1344

Constraints

$2 \leq N \leq 7$

Time Limit:

Maximum time allowed per test case is 5 seconds

Scoring:

20 points for each correct output.

0 points for each incorrect output.



Old Mutual - CSSA Computer Olympiad Training Day 1



IOIWari board game (IOI 2001)

Submitted by: Graham

Description

The Mancala family of games with beads and pits is among the oldest forms of human entertainment. This task introduces a version of the game especially developed for the IOI. Two players play the game on a round board with seven pits around the edge. In addition, there is a bank for each player. The game begins by randomly distributing 20 beads into the pits so that each pit contains at least 2 and at most 4 beads. The two players move alternately. To move, the player chooses a non-empty pit and takes all beads out of the pit, and holds them in her hand. As long as there are beads in the player's hand, she considers the pits in clockwise order, starting one after the emptied one, and performs the following operations:

More than one bead in your hand: If the current pit already contains 5 beads, then take one bead out of the current pit and place it into your bank, otherwise place one bead from your hand into the current pit.

- One bead in your hand: If the current pit contains at least one and at most four beads then move all beads from the pit and the one from your hand into your bank, otherwise (the pit contains 0 or 5 beads) place the bead in your hand into the opponent's bank.
- The game is over when, after a move, all pits are empty. The winner is the player with most beads in her bank. The starting player always has a winning strategy. You are to write a program, which plays Ioiwari as the starting player and wins. The evaluation opponent plays optimally, that is, once given a chance, it will win and your program will lose.

Task

What must be done in clear unambiguous language.

Input and Output

Your program reads input from standard input and writes output to standard output. Your program is player 1, and the opponent is player 2. When your program is started, it must first read a line with 7 integers p_1, \dots, p_7 : the initial number of beads in pits 1, ..7, respectively. The pits are labeled with integers from 1 to 7 in clockwise direction on the board. After this, the game starts with empty banks. Your program should play as follows:

- If it is your program's turn to move, then your program should write the label of the pit describing the move to standard output
- If it is your program's opponent's turn to move, then your program should read the label of the pit defining the move (the pit from which the beads are removed) from standard input.

Tools

You are given a program (`ioiwari2` on Linux, `ioiwari2.exe` on Windows), which plays from one initial game position optimally as Player 2. It will first write to standard output the first line your program is supposed to read, describing the initial values of beads in that game: 4 3 2 4 2 3 2

After this, the program will play the game, trying to read Player 1's moves from standard input and writing its own moves to standard output. You can run your program and `ioiwari2` in separate windows and transfer the conversation manually to both programs. `ioiwari2` records the dialogue in the file `ioiwari.out`.

Programming Instructions

In the examples below, you are reading the last integer of the input into variable `last` and the variable `mymove` contains your move.

If your program in C++ and use `iostreams`, you should use the following implementation for reading standard input and writing to standard output:

```
cout<<mymove<<endl<<flush;
cin>>last;
```

If your program in C or C++ and use `scanf` and `printf`, you should use the following implementation for reading standard input and writing to standard output:

```
printf("%d\n", mymove);
fflush (stdout);
scanf ("%d", &last);
```

If your program in Pascal, you should use the following implementation of reading standard input and writing to standard output:

```
Writeln(mymove);
Readln(last);
```

Example

Here is a correct sequence of 6 moves

	Pit and bank contents after each move								
Operation/Pit label	1.	2.	3.	4.	5.	6.	7.	Bank1	Bank2
Initial situation	4	3	2	4	2	3	2	0	0
Player 1's move: 2	4	0	3	5	0	3	2	3	0
Player 2's move: 3	4	0	0	4	1	4	0	3	4
Player 1's move: 5	4	0	0	4	0	0	0	8	4
Player 2's move: 4	0	0	0	0	1	1	1	8	9
Player 1's move: 5	0	0	0	0	0	0	1	10	9
Player 2's move: 7	0	0	0	0	0	0	0	11	9

Scoring

If your program wins a test run, then you get 4 points for that test, a tie in a test gives you 2 points for that test, and otherwise you get 0 points for a test.