Kakuro

Submission deadline:	**2011-12-31 23:59:59**	446337.070 sec
Evaluation:	**0.0000**	
Max. assessment:	**5.0000** (Without bonus points)	
Submissions:	0 / 50 Free retries + 10 Penalized retries (-2 % penalty each retry)	
Advices:	0 / 0	

The task is to develop a program which solves kakuro puzzle.

The input of the program is problem definition. The kakuro is given in a form of a rectangular grid. One row of the grid is represented by exactly one input line. The individual cells are separated by spaces (at least one space is between two consecutive cells). A cell may have the form:

X
: represents an unused (black) cell in the grid,

.
: represents an empty cell in the grid (a cell which is to be filled by a number),

num1\num2
: represents a cell that defines the sums. The consecutive sequence of empty cells below shall have sum of `num1` and the consecutive sequence of cells to the right shall have sum of `num2`,

num1\X
: is a cell like the above, except that only the vertical sum is provided,

x\num2
: is a cell like above, except that only the horizontal sum is provided.

The output of the program is the description of the solution. If there is exactly one solution of the input kakuro, the program shall print it. The format is shown below. Otherwise the program shall display the total number of unique solutions (i.e. none, or more than one solution).

The program must sanitize input data. If an invalid input is given, the program shall display an error message (see below) and terminate. The following is treated as an error:

- cell content different from the five variants above,
- the shape of the input grid is not a rectangle (i.e. rows with different number of cells),
- there is not a cell with vertical sum above a consecutive sequence of vertical empty cells,
- there is not a cell with horizontal sum to the left of a consecutive sequence of horizontal empty cells,
- a cell defines the sum, however, there are not empty cells in the corresponding direction,
- there is a sequence of empty cell of length 10 or more,
- the sum is less than 1 or more than 45,
- the input grid exceeds 32x32 (this is the limit of the problem size).

A simple introduction to kakuro game: the goal is to fill-in empty cells in the input grid. All empty cells must be filled by numbers 1-9. There are sums defined for horizontal/vertical sequences of consecutive empty cells, the filled values must match the sums. Moreover, each number (1-9) may occur at most once in a consecutive sequence of horizontal/vertical cells.

The resulting program is somewhat more complex than a regular homework. It is a contest problem, designed for students - experienced programmers - bored by the simplicity of the regular homework. It is designed for those who seek a challenge. The submitted program will be evaluated in two stages. First, if the program succeeds in the mandatory tests, the program will be awarded a nominal evaluation. Next, if the program is optimal, it may be awarded further points:

- 5 points if the overall score is more than 75% of the reference solution score,
- 5 points if the overall score is among the top 10% of the submitted solutions (the 10% is faculty-wide and includes Czech groups as well).

The additional 5+5 points will be awarded after the deadline of the contest homework. There is no advice available for the contest

homework.

The program will be tested as usually, in a limited environment with time and memory limits set. The exact limitations are listed in the reference solution test log. The program must be optimized to succeed in the contest test. The contest input problems are kakuros with multiple solutions, there is up to 100000 solutions to be tested in the time limit.

Sample program run:

```
Enter the kakuro:
X     X     16\X  16\X  X     X     15\X  16\X  X     X     X     X
X     X\8   .     .     7\X   X\13  .     .     X     7\X   12\X  X
X     16\17 .     .     .     29\9  .     .     19\15 .     .     X
X\9   .     .     6\23  .     .     .     11\11 .     .     .     X
X\16  .     .     .     23\19 .     .     .     .     9\X   X     X
X     X     X\22  .     .     .     19\8  .     .     .     30\X  6\X
X     X     6\X   15\21 .     .     .     .     11\19 .     .     .
X     X\22  .     .     .     11\21 .     .     .     4\8   .     .
X     X\7   .     .     X\3   .     .     X\9   .     .     .     X
X     X     X     X     X\12  .     .     X     X\12  .     .     X
One solution:
X     X     16\X  16\X  X     X     15\X  16\X  X     X     X     X
X     X\8   1     7     7\X   X\13  4     9     X     7\X   12\X  X
X     16\17 7     9     1     29\9  2     7     19\15 6     9     X
X\9   7     2     6\23  6     9     8     11\11 7     1     3     X
X\16  9     6     1     23\19 7     1     3     8     9\X   X     X
X     X     X\22  5     9     8     19\8  1     4     3     30\X  6\X
X     X     6\X   15\21 6     5     8     2     11\19 6     8     5
X     X\22  5     9     8     11\21 7     5     9     4\8   7     1
X     X\7   1     6     X\3   2     1     X\9   2     1     6     X
X     X     X     X     X\12  9     3     X     X\12  3     9     X
```

```
Enter the kakuro:
X     5\X   7\X   9\X
X\6   .     .     .
X\15  .     .     .
Total solutions: 4
```

```
Enter the kakuro:
X     3\X   5\X
X\3   .     .
X\4   .     .
No solution.
```

```
Enter the kakuro:
X     3\X   5\X
X\3   .
X\5   .     .
Invalid input.
```

Sample data:	**Download**

Additional bonus points may be assigned in this task. The bonus depends on contest score. The score is computed from submission quality (reliability and speed), compared to the reference solution quality and/or to the quality of solutions submitted by other students.	**Results**
Contest score:	-- (0.00 % of the reference solution)
Position:	--
Bonus:	-- (incomplete, contest still running)

Submit: [redacted] **Submit**

Reference

- **Evaluator: computer**
  - Program compiled
  - Test 'Základní test s parametry podle ukázek': success
    - result: 100.00 %, required: 100.00 %
    - Total run time: 0.022 s
    - Max. run time: 0.005 s (limit: 2.000 s)
    - Peak mem usage: 11776 KiB (limit: 18434 KiB)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Test mezních hodnot': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 0.027 s
    - Max. run time: 0.006 s (limit: 2.000 s)
    - Peak mem usage: 11776 KiB (limit: 18434 KiB)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Test ošetření nesprávných vstupů': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 0.048 s
    - Max. run time: 0.005 s (limit: 2.000 s)
    - Peak mem usage: 11776 KiB (limit: 18434 KiB)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Test náhodnými vstupy (jedno řešení)': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 0.066 s
    - Max. run time: 0.006 s (limit: 2.000 s)
    - Peak mem usage: 11872 KiB (limit: 18434 KiB)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Test náhodnými vstupy + kontrola práce s pamětí': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 0.203 s
    - Max. run time: 0.022 s (limit: 2.000 s)
    - Peak mem usage: 18736 KiB (limit: 21363 KiB)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Test náhodnými vstupy (více řešení)': success
    - result: 100.00 %, required: 50.00 %
    - Total run time: 0.113 s
    - Max. run time: 0.060 s (limit: 2.000 s)
    - Peak mem usage: 11872 KiB (limit: 21363 KiB)
    - Mandatory test success, evaluation: 100.00 %
  - Test 'Soutěžní test':
    - Total run time: 2.370 s
    - Max. run time: 0.919 s (limit: 5.000 s)
    - Peak mem usage: 11932 KiB (limit: 21363 KiB)
    - Sum time (correct answers): 2.370 s
    - Penalty time (wrong answers): 0.000 s
    - Cyclomatic complexity: 17
    - Contest score: 2746.020 pt (= 3000.0 * 1.0000 - 100.0 * 2.370 - 1.0 * 17.000)
  - No memory block leaks detected - ok.
  - Overall ratio: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00 * 1.00 * 1.00)
  - Overall contest score: 2746.020
- Total percent: 100.00 %
- Total points: 1.00 * 5.00 = 5.00

Total	Average	Maximum	Function name

SW metrics:	Functions:	**26**	--	--	--
	Lines of code:	**719**	**27.65 ± 14.78**	**49**	`printCell(TCELL *,int,int)`
	Cyclomatic complexity:	**254**	**9.77 ± 5.02**	**17**	`prepareRuns(TKAKURO *)`