Note

Émetteur(s) :	Sopra Stéria Battle Team 2016
Destinataire(s) :	Collaborateurs Sopra Stéria
Copie(s) :	
Objet :	**Exercices Battle Code 2016**

0. Introduction

The subjects described below can be performed in any order and are not interdependent.

Your goal is to find relevant solutions to solve the given problems.

Each exercise is made of 3 questions (with more and more complex datasets). Theoretically, all 3 questions do not require any modification on the algorithm. If it is well written, question 3 should be a formality.

Although the expected format is clearly stated in each statement, it is reminded that all requested answers are in international format (eg 01d02h15m45s).

Question 3 generally contains a large data set to test the reliability of the algorithms, so it may be contained in a text file. It is not necessary for the program to read the file. A simple copy and paste of the contents into a variable is allowed.

Be vigilant, the virtual referee does not think, so talk to him in the format he expects from you! There is only one correct answer per question. If the response is a list, it must be ordered as specified in the statement.

Please read the rules of each exercise to give the answer in the expected format. Before giving an answer, remember to read the statement to get sure of the format as well as conventions adopted (direction of X and Y axes, index of the first cell ...)!

In the examples, the expected responses (i.e. with the correct format) are green.

Line breaks (\n and \r) will be automatically deleted from the answer given to the virtual referee. They are therefore optional (exercises 7 and 16).

Be careful when sending the source code. It should be self-sufficient, it is intended for the validation team to run it on their own computer.

The code must be made to run on another machine. Provide any used custom libraries.

You may add a readme.txt file with your sources, to help the validation team to execute your code.

Make sure you send the correct source code.

The validation team will only test the 3 questions provided in each statement.

If the source code used by the validation team does not resolve the statement, your points on the exercise will be canceled.

The easiest exercises are estimated at 10-15 minutes. The most difficult exercises are estimated at 45-60 minutes.

These durations are indicative only, the Battle Code team is not responsible for the developers' performances.

It's up to you to make your own strategy!

The more difficult the exercise is, the more profitable it is, but in case of error, debugging is much more time consuming.

Time to get Serious! Here are the statements, sorted in ascending order of difficulty.

A bonus is awarded to the first team (World-wide) that solves the exercise.

A penalty is applied for each wrong answer (cumulative effect). Once per submit.

To score points, you must correctly answer all three questions, and provide the source code.

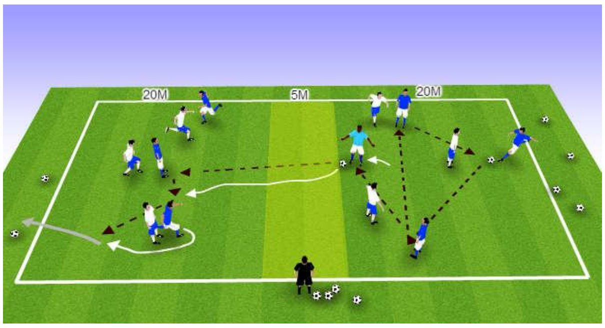
Numéro	Exercice	Points	Bonus	Malus
1	Football game	5	2	1
2	Encrypted message 1	6	2	1
3	Boby Lapointe	6	2	1
4	Counting M&M's	7	2	1
5	Encrypted message 2	7	2	1
6	Sum of a clock	7	2	1
7	Minesweeper generator	8	3	1
8	Encrypted message 3	10	3	1
9	Exary language	10	3	1
10	House of cards	15	5	1
11	The good and the bad developer	20	7	2
12	Knight on the night	25	8	2
13	Maze reader	25	8	2
14	Connect 4	25	8	2
15	Unpaid invoices	35	12	2
16	Building an automated Lego Pick-up	45	15	2
17	Battle Code Rocks language	50	18	2

Get your keyboards ! Ready ? Code !

1. Football game (5pts +2/-1)

Introduction

An amateur football team tries to estimate its chances of winning according to the selected team. This is a classic football match. Not a Rugby match, not an American football, nor a gaelic football match... Not saying they are not interesting, but it would have been more complicated. At classic football, there is a ball, you shoot, you goal, you score. That’s it. Any kid can do that. For fun. For free...



Rules

- Our team scores a goal against the opponent every X minutes. The first goal was scored in the Xth minute.
- Our team scores an own goal every Y minutes. The first goal was scored in the Yth minute.
- The opposing team is a professional team, she doesn’t score any own goal.
- The opposing team scores goals every Z minutes. The first goal was scored in the Zth minute.
- The expected answer is the final score after 90 minutes.
    - If our team scores three goals in the opposing cages, 1 goal in our own cages, and the opponent scores 3 goals in our cages, the expected answer is **3-4**

Questions

What is the score with those teams:

1. Our team scores 1 goal every 35 minutes in the opposing cages
   Our team scores 1 goal every 65 minutes in our own cages
   The opponent scores 1 goal every 35 minutes in our cages

2. Our team scores 1 goal every 15 minutes in the opposing cages
   Our team scores 1 goal every 30 minutes in our own cages
   The opponent scores 1 goal every 10 minutes in our cages.

3. Our team scores 1 goal 1 minute into the opposing cages
   Our team scores 1 goal every 120 minutes in our own cages
   The opponent scores 1 goal every 2 minutes in our cages.

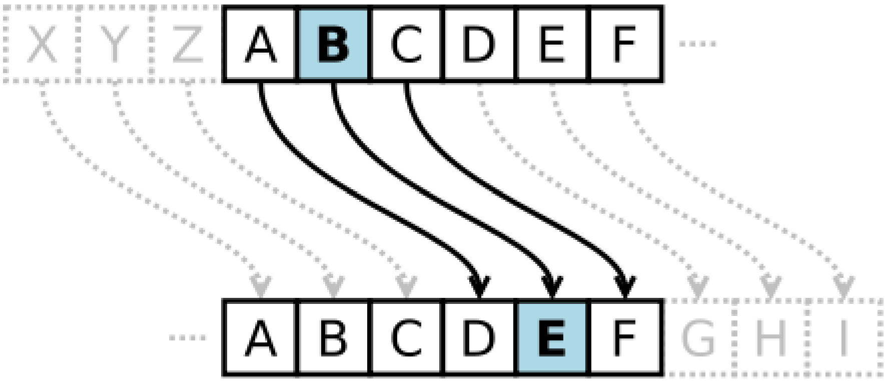
2. Encrypted message 1 (6pts +2/-1)

Introduction

A very simple method to encrypt a message is the Caesar Cipher. It was used more than 2000 years ago, ensuring that the message cannot be intercepted. The principle is simple: shifting the alphabet letters by a fixed number. Today this method is not reliable anymore because it became famous, and if you are able to guess a letter (by frequency analysis), you are able to crack the code. You may even test the 25 possibilities until you see an intelligible message.

Rules

- The number of Caesar is provided in the question.
- If the number of Caesar is “+1”, then clear-text “A” becomes “B” in the encoded message



In the image above, the number of Caesar is "+3", so clear-text "B" becomes "E" in the encoded message.

- The purpose of the exercise is to decipher a coded message.
- Messages are composed only of the 26 capital letters of the Latin alphabet.
  - Messages contain no accent.
  - White spaces and special characters are not encoded

Example

- If the number of Caesar is « +5 », the encrypted message « MJQQT BTWQI ! » becomes « **HELLO WORLD !** »

Questions

Please decipher the following messages:

1. Number of Caesar: +1. Message: DBFTBS DJQIFS
2. Number of Caesar: -3. Message: QEFP FP X KBDXQFSB HBV
3. Number of Caesar: +404. Message: TUZF: GEQ YAPGXA FIQZFK EUJ FA PQOAPQ FTUE AZQ !

3. Boby Lapointe (6pts +2/-1)

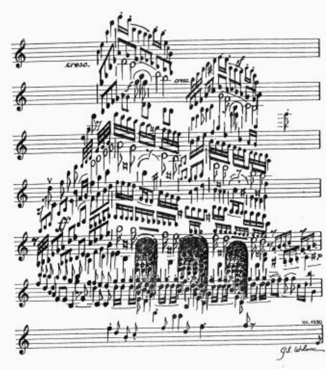
Introduction

Boby Lapointe (1922-1972) was a French singer, but was also a researcher and an early computer scientist.
He invented the "bibi-bit" system that uses syllables instead of letters and numbers to code in hexadecimal numbers.
The system has been patented, we will modify it to use the base 10, will be called "dodo-deci".
The exercise is to write a converter dodo-deci → Decimal.

Correspondance tables

Each digit becomes a syllable, using the following mapping:

decimal	dodo
0	po
1	pa
2	pe
3	pi
4	to
5	ta
6	te
7	ti
8	ko
9	ka



To use the "dodo-deca", he also added the, following operators:

operator	dodo
+	do
-	da
*	de

Rules

- Conventional operations priorities are used, that is to say that multiplication takes precedence over addition and subtraction.
    - padopedepidape o = 1 + 2 * 3-2 = 1+ (2 * 3) -2 = 5
- The first character provided cannot be a "d", that is to say, the operation necessarily begins with a positive number.
- An operator is ALWAYS surrounded by two digits. It is not possible to find two successive operators.
    - There is no need to take care of "++", "+ -", "- +", "*+" and "*-"
- There will only be integers.

Example

Without operators	Question	pepopate
	Result	**2016**

With operators	Question	todopedepipedapapa
	Decimal	4+2*32-11
	Result	**57**

Questions

1. papopeto
2. papetedotidepetadokadopatekadatatidopipidopetidetotidatitopidotepetadapipeta
3. papetedokokopadotidepetadapatadepekodepadekadopatekadotatipipidopetidetotidatitopidotepetadotokadetadekodepodetotidapipetadopepadekopedopipokadakopopedatatopipidotitedopo

4. Counting M&M's (7pts +2/-1)

Introduction

When buying M&M's, you must be sure of what you buy. If your favorites are the blue ones, you don't want it to be a minority! Wanna check it out together?

Rules

- Everything starts with an empty pack.
- The input is a sequence of "C NNN" separated by commas.
  - "C" is the M&M's color (R, G, B, Y, O)
  - "NNN" is the number of M&M's added (from 000 to 999)
- Several lines can have the same color
- The purpose is to give the color of M&M's majority.

Example

Input : R 005,R 010,Y 090,G 020,B 010,R 008,Y 090,O 170
The expected answer is **Y**, because it is the color of M & M's has the most (180 M&M's)

Questions

What color has the majority in these packs?

1. O 018,Y 028,R 097,G 078,O 071,O 045,B 037,O 055,Y 005,R 054

2. R 204,B 948,Y 077,Y 937,O 247,O 478,O 285,Y 194,O 466,B 922,G 067,B 315,R 597,B 293,O 995,R 477,Y 716,R 965,G 770,R 227
3. B 828,R 688,R 867,R 810,R 720,R 236,R 223,R 900,G 544,G 312,G 879,B 616,B 646,B 898,R 155,R 255,G 296,G 469,Y 000,G 726,G 347,G 330,B 436,B 139,B 129,B 395,B 687,B 844,R 601,R 524,R 548,R 238,R 231,G 251,G 013,G 664,G 928,G 996,B 598,B 855,B 354,B 926,B 645,B 771,B 188,B 025,B 437,B 323,R 419,R 573,R 948,R 925,R 279,R 776,R 601,R 369,R 892,R 742,R 839,R 967,G 246,G 543,G 890,G 454,G 577,G 563,G 548,G 595,G 758,G 869,G 313,G 989,B 786,B 806,B 755,B 017,B 965,B 548,B 414,R 423,G 877,B 815,R 106,G 477,B 728,R 488,G 969,B 209,R 177,G 315,B 091,R 660,G 226,B 462,G 064,R 644,G 274,B 096,R 314,G 966,B 933,B 731,B 542,G 584,R 683,R 571,R 126,G 666,B 978,B 535,G 827,G 031,G 889,B 213,B 636,B 924,R 873,R 465,R 122,R 603,R 769,G 860,B 100,G 706,R 616,G 713,B 989,R 869,G 378,B 899,G 681,R 553,G 178,B 191,R 289,G 567,B 126,R 326,G 587,B 234,R 351,G 524,B 516,R 692,G 311,B 677,R 580,R 274,R 985,R 926,R 488,R 462,G 044,G 842,G 868,G 357,G 588,G 943,G 426,B 157,B 542,B 743,B 361,B 599,B 311,G 075,R 403,B 835,G 789,R 894,G 310,B 709,R 301,G 273,G 165,G 919,G 589,B 836,B 321,R 007,R 431,R 308,R 335,G 667,G 976,G 445,B 738,B 332,B 407,B 019,R 983,R 507,R 009,R 115,R 408,B 818,B 247,B 432,B 575,R 252,R 942

5. Encrypted message 2 (7pts +2/-1)

Introduction

The Caesar Cipher is too famous, let’s make a new encryption method: The alphabetically-not-aligned-that-depends-on-the-previous-letter-which-prevents-any-frequency-analysis-and-is-therefore-impossible-to-decipher-unless-you-know-the-method-of-encryption. Let's go!

Rules

- The first letter of the coded message is clear (i.e. not encrypted).
- The next letter is shifted of the amount of the previous letter
    - If the first letter is A, the second will be shifted by one notch
    - If the third letter of the clear message is "E", then the fourth letter will be shifted by 5 notches
- The purpose of the exercise is to decipher a coded message.
- Messages are composed only of the 26 capital letters of the Latin alphabet.
    - They contain no accent.
    - White-spaces and special characters are not encoded.

Example

- If you encode « **SOPRA STERIA!** » you get « SHXPQ JDIAJK! »
    - S is the 19th letter ➔ 19+0 = **19** ➔ « S »
    - O is the 15th letter ➔ 15+**19** = 34 ➔ 34-26 = **8** ➔ « H »
    - P is the 16th letter ➔ 16+**8** = **24** ➔ « X »
    - R is the 18th letter ➔ 18+**24** = 42 ➔ 42-26 = 16 ➔ « P »
    - Etc...

Questions

Please decipher the following messages:

```
1. HMEJ GL SH!
2. YNI BJKWI WLF VWPI...!!!
3. YXWLAPETOJEZU  ...  NVWIU IXMBQFZT   ...  JZABCDEFGHIJKDWPIB...
```

6. Sum of a clock (7pts +2/-1)

Introduction

Take a digital clock that displays the hours and minutes (24h format). What will happen if we make the sum of the hours and minutes all over a period?

Rules

- The period cannot exceed 24 hours.
- At 8:53, the sum of the hours and minutes is **61**

Example

- From 15:15. (included) to 15:18. (not included), the amount of hours and minutes is 30 (15:15) + 31 (15:16) 32 (15:17) = **93**

Questions

What is the sum on these periods?

1. 00:00 to 01:00
2. 09:59 to 23:30
3. 15:00 to 14:58

7. Minesweeper generator (8pts +3/-1)

Introduction

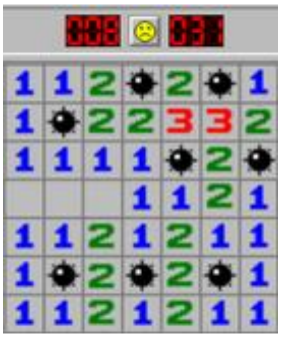
Everyone knows the famous minesweeper on Windows, where the goal is to find all the mines without walking on one. When walking on a box, the number displayed is the number of mines around the box. Here we go to generate a minesweeper board !

Rules

- The board is a square with sides of length N
- The input data are the coordinates of every mines
    - A whitespace separates the X and Y coordinates of a mine
    - A comma separates each mine
- The top left cell is the cell (1,1)
- The expected answer is the full board.
    - A digit indicating the number of neighboring mines
    - An « x » on a mine

Example

On the following 7x7 board :



4 1,6 1,2 2,5 3,7 3,2 6,4 6,6 6

The expected answer is:

```
112x2x1
1x22332
1111x2x
0001121
1121211
1x2x2x1
1121211
```

Questions

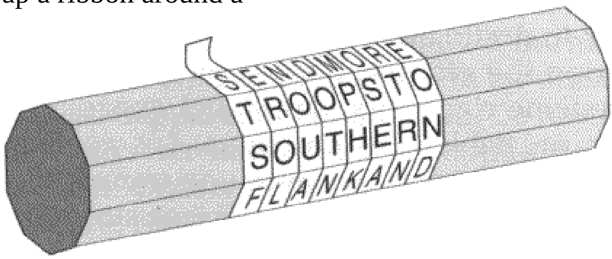
Write a program that generates the following boards:

1. 5x5 board: 5 5,2 3,1 1,3 3,2 5
2. 10x10 : 8 7,10 3,10 2,3 5,4 8,9 5,3 2,4 10,9 7,1 8,4 3,5 2,2 10,5 7,4 9,6 2,2 1,6 3,3 10,6 10
3. 100x100 : Minesweeper_q3.txt

8. Encrypted message 3 (10pts +3/-1)

Introduction

Among the Spartans, the scytale, also known under the name stick of Plutarch, was a wooden stick used for reading or writing an encrypted dispatch. The idea is to wrap a ribbon around a stick to read to read the message.



Rules

- The first letter of the coded message is the first letter of final message.

- In the question, the diameter of the stick to use will be provided, and the height of each letter
    - The number of letters to wrap once around the stick will be an integer. If the result is 9.023 letters, it will be rounded to 9.
- The format of the question is "XX,YY,MESSAGE".
    - XX is the diameter of the stick in centimeters
    - YY is the height of a letter in millimeters
- The purpose of the exercise is to decipher a coded message.
- Messages are composed only of the 26 capital letters of the Latin alphabet.
    - They contain no accent, no white-space, and no punctuation.
- The message completely covers the stick, there is no white characters to add.
    - If there is room for 10 characters per wrap around the stick, the message will be a multiple of 10 characters.

Example

- If one decodes “STSFEROLNOUADOTNMPHKOSEARTRNEOND" 1,4cm diameter stick, letters of 1cm height, we get 4 letters per wrap around the stick.
  The message is 32 characters, which implies 8 times around the stick. Is obtained "**SENDMORETROOPSTOSOUTHERNFLANKAND**".

Questions

Please decipher the following messages:

1. 13,102,LIUTUMRHKYFEEOAR
2. 28,110,HHHPLOLWMAEEOFAAMTAECADLMIIDINELMSRVTUNOWTSEYNSW
3. 18,51,SODRNDULTRTOVSADILEYAIPIETBNTXASORDRIUGRANTNAEVOSPANDETSS
   INIRNDPRETOCSNISCEICENEOEVFRRAHREOFSAOIFENIOFTSTRTONOAFFWPEMIRA
   LETEARAACMBOUHRRONTAALGREIECDILNEYOMNDEPOBCSYPOGESUNUEIEESSVSBP
   SSTSATIESLRIESTNCNLEIONRCHLOTORCGEVLIEMHPVORSIISAPEMIRASCEIDRME
   CGMCENSEEANEAMHSTTRHRTSNEAASHIEKISISLDTENNENOSTLDOADSTFPAHEEMNI
   ICRRTANDASGVOAAITGVKWIENSSOAEAEETPSTTNDSLTRAOURESDCUHOLRLURTROE
   EFTTTCIOEMABTRFITADSBNEHAONUIESIDSENLGRSLTNITTSISETIHINUHFOYMRV
   ENNSIOSSAUEIGOERROTNSRRHVODMFEATSMIAFQAEMGEUOGTIUTNSEDCSHINEIDI
   MBCTQOFSOTNEYECUNOTNOTNLSOASRIPEETESMLOMORNGAAFPIPAN

9. Exary language (10pts +3/-1)

Introduction

Binary is good, but anyone can read it. A cryptologist has invented *exary* language

Rules

- The input message is made up of ASCII characters, which have to be translated into 7bits binaries
- The output encoded message is made up of x blocks
- A block is separated from another by a white space
- Two consecutive blocks used to produce a series of bits of the same value (only 1's or only 0's in binary):
    - First block (header): it is always x or xx. If it is x, the next series will contains 1. Otherwise, it will contain 0.
    - Second block (bits): The number of x in this block is the number of bits in the series

Example

Let's take a simple example with a single character message: capital C. C in binary is 01000011 on 8 bits, so 1000011 on 7 bits, which gives in *exary*:

- x x (the first series contains a single 1)
- xx xxxx (the second series contains four 0)
- x xx (the third series contains two 1)

C is therefore: **x x xx xxxx x xx**

Second example, we want to encode the « B C » (the 21 following bits 1000010_0100000_1000011):

- x x (only one 1)
- xx xxxx (four 0)
- x x (one 1)
- xx xx (two 0)
- x x (one 1)
- xx xxxxx (five 0)
- x x (one 1)
- xx xxxx (four 0)
- x xx (two 1)

« B C » is therefore: **x x xx xxxx x x xx xx x x xx xxxxx x x xx xxxx x xx**

Questions

Write a program which translates those messages into *exary*:

1. BC
2. battle code rocks
3. Chuck N0rr1s' k3yb0@rd h@s 2 k3ys: x @nd wh1t3 sp@c3.

10. House of cards (15pts +5/-1)

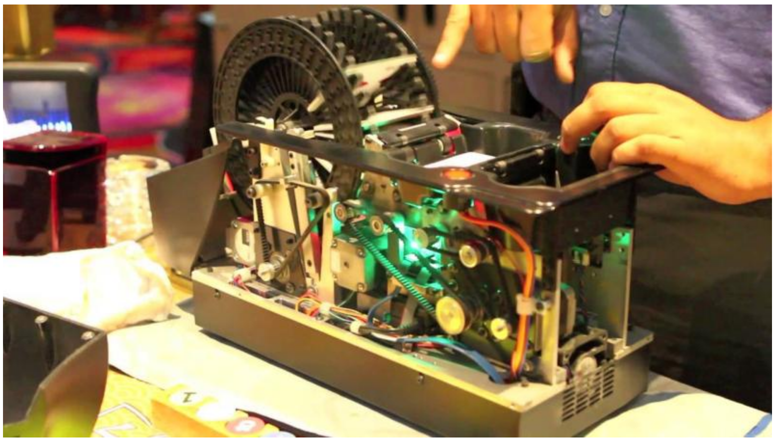
Introduction

A Startup wants to create an automatic card shuffler and has developed an Alpha prototype release which gives a "random" card out of a deck. Since randomness does not really exist, it appears that knowing the input deck, one can predict what card will go out of the shuffler.

It is proposed to guess the output card, when a sorted deck is provided as input.

Rules

- The cards follow the order value in poker: ACE, KING, QUEEN, JACK, 10, 9, 8, 7, 6, 5, 4, 3, 2; and the color order: SPADES, HEARTS, DIAMONDS and CLUBS.
- These cards are numbered from 1 to N, with N the number of cards provided. Card #1 being on top, then the card #2 ... until the card N. The first card of a pack is the ACE OF SPADE and the last 2 OF CLUBS.
- The shuffler takes the top card (**#1 = ACE OF SPADES**), replaces it below the deck, throws away the next one (**#2 = KING OF SPADES**), places the #3 below the deck, throws away the #4 etc...
- The output card is the last remaining card.
- The expected response is the card number in the initial bunch, its color, and the value of the card:
  - Values: AKQJT98765432.
    - **Note the "10" is represented as a "T"**
  - Color: SHDC (for Spades, Hearts, Diamonds and Clubs)
  - Card number: number between 1 and N, where N is the size of the initial deck.
- The algorithm has to be efficient enough to sort millions of cards.



Example

- 39 packs of 52 cards outputs **7D2009** matches 7 of Diamond, who was at position 2009 in the initial deck.
- 18 packs of 52 cards outputs **TH849** matches 10 of Heart, who was at position 849 in the initial deck.

Questions

Write a program that will find the value of the remaining card (value, color and rank in the original deck) by the number of packs of 52 cards.

1. We input a deck of 52 playing cards representing 1 pack.
2. We input a deck of 2028 playing cards representing 39 packs of 52 cards.
3. We input a deck of 5 200 000 playing cards, representing 100 000 packs of 52 cards

11. The good and the bad developer (20pts +7/-2)

Introduction

A customer wants a GUI to monitor and administer a robot that will soon be put on the market. A team is in charge of development. The team is not dedicated full time, several developers are involved, which hardens the task. Of course, the team is made of good and bad developers.

The good developer, he sees a page to code, well, he codes it. This is a good developer. The bad developer, he sees a page to code, well, he codes it. But it's a bad developer. It's different.

Rules

- A good developer is able to create X_g complete pages in Y_g minutes
- A bad developer corrupts X_b complete pages in Y_b minutes, slowing down the good developer.
- A developer works 8 hours a day.
- All developers are working at the same time.
- The purpose is to calculate how much time is required for the pages to be ready.
- The result's format is expected to be **00d01h20m30s**,
- The result is truncated to the second.
  - If the result is 10.84 seconds, the expected response is **00d00h00m10s**
  - If the wrong developers completely prevent good developers to work, the expected response is **FAIL**

Examples

First example: There are only two developers for one page. The good developer makes a page in one day (1d/page), and the bad developer corrupts a page in four days (4d/page).
Short version: 1 page to develop, 1 good developer (1d/page), 1 bad developer (4d/page)
The page will be ready in 1.3333333 days = 1 day, 2 hours and 40 minutes.
The expected answer is **01d02h40m00s**

Second example: A team of three developers must develop 5 pages. The first one is a good developer, he makes a page per day. The second one is a bad developer, he corrupts a page in 2 days. The third one is a bad developer, he corrupts a page in 3 days.

The 5 pages will be ready in 30 days.
So the expected answer is **30d00h00m00s**

Questions

Write a program which computes the time required to developer to finish the job*:*

1. 1 page to develop, 1 good developer (1d/page), 1 bad developer (3d/page)
2. 3 pages to develop, 3 good developers (2d/page, 3d/page, 6d/page), 1 very bad developer (1d/page)
3. 4 pages to develop, 3 good developers (2d/page, 3d/page, 6d/page), 2 bad developers (5d/2pages, 10d/3pages)

12. Knight on the night (25pts +8/-2)

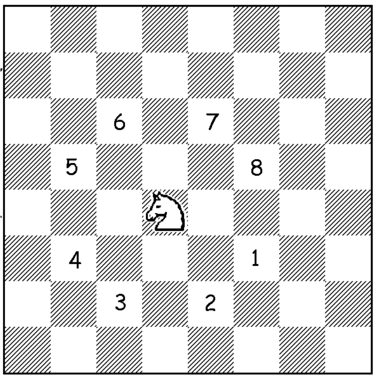
Introduction

Breaking news ! A knight escaped this afternoon from his board and is walking freely. He is white, has a rather long mane, brown eyes, a horse's breath, moves in "L", and was seen for the last time next to the North West hors'pital but is often seen at the South West King's Corner.

Rules

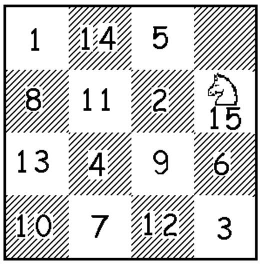
The instructions are:

- The knight is on a chessboard of size NxM (provided in the question)
  - N columns, M rows
- A square is a hay bar.
- The knight must move from a hay bar every 15 minutes not to be spotted.
- The knight starts his travel in the top left corner. (near the hors'pital)
- The knight moves in "L" in one of 8 possible bars (see figure)
- The knight will always prefer to go to "1". If this is not possible, it will try to go to 2. If it is still not possible, it will try to go to 3, etc ...
- The knight can't go to a hay bar he has already visited (it would be noticed)
- The run is completed when the knight has no directly reachable hay bar.
- The expected response is the duration (in minutes) of the knight run (including the first and the last squares).



Example

On a 4x4 board, the knight can go in 15 different hay bars. His run therefore lasted 225 minutes. The expected answer is: **225**



1	14	5	
8	11	2	15
13	4	9	6
10	7	12	3

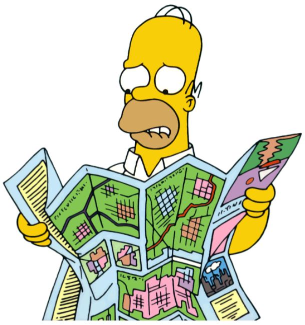
Questions

Write a program which calculates the elapsed time of the knight's run through the following boards:

1. 5x5 board
2. 10x10
3. 1000x1500

13. Maze reader (25pts +8/-2)

Introduction



Homer found the plans of his cellar. Help him out to read it!

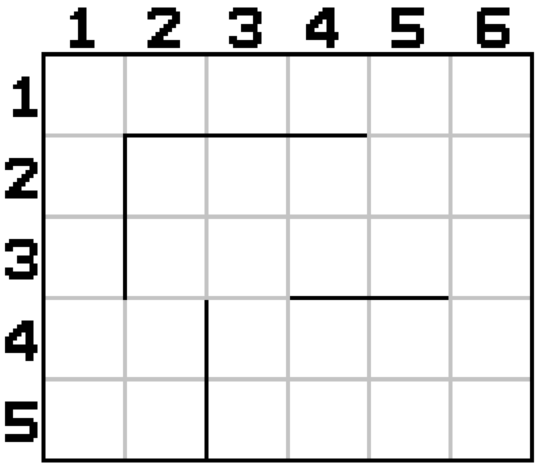
Rules

- For an NxM (N columns, M rows) maze, the input contains a list of walls specifying coordinates of a cell, and the direction blocked by the wall
- The first character gives the direction blocked by the wall (Left, Right, Up, Down).
  - L 2 3 => there is a wall to the left of the cell (2,3) → there is a wall between cells (2,3) and (1,3)
- The goal is to count the number of walls on each cell.
- The expected answer is the number of cells having x walls (where x is 0,1,2,3 or 4).
- The first coordinate is X, the second is Y.
- The top left cell is the cell (1,1)
- The edge of the maze is a wall

Example

Let the following 6x5 maze:

```
R 1 2
L 2 3
D 2 1
U 3 2
D 4 1
L 3 4
R 2 5
D 4 3
U 5 4
```



The number of cells having only 1 wall is **16**

Questions

Write a program that counts the number of similar cases::

1. On this 4x3 board, give the number of cells having 0 walls:
   U 4 3
   R 2 3
   L 2 1
   L 3 1
   D 2 2
   D 3 1
2. On the 30x10 board from the file Labyrinth_q2.txt, give the number of cells having 1 wall
3. On the 50x150 board from the file Labyrinth_q3.txt, give the number of cells having 2 wall

14. Connect 4 (25pts +8/-2)

Introduction

“Connect 4” is a 2-players game, where the goal is to be the first player to get four of his colored checkers in a row (horizontally, vertically or diagonally). At each player’s turn, he inserts a checker at top of a column, which slides to the last free space. Here we want to analyze a board to determine where are the winning checkers.

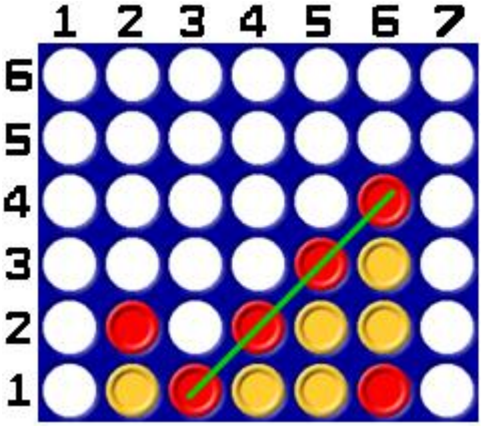
Rules

- The board size is R×C (R rows and C columns)
- The input data is the column numbers of each move
  - Each move is separated with a comma
  - Players alternate turns after playing a checker.
  - At each move, a player inserts a checker in the announced column. The checker slides to the last empty line of this column.
- The bottom left cell position is (1,1)
- The expected result is the checker from the winning group, in the format "**X Y**", which follows the rules:
  - SW⇔NE, the requested coordinate is the bottom left checker
  - W⇔E, the requested coordinate is the left checker
  - NW⇔SE, the requested coordinate is the top left checker
  - N⇔S, the requested coordinate is the top checker
- The winning move is not necessarily the last move
- There is only one winning combination

Example

On this 6x7 board :
4,3,5,4,5,6,2,2,6,5,6,6

The checkers making the winning group are:
(3,1),(4,2),(5,3),(6,4)
The configuration of checkers is SW⇔NE so this is the bottom left checker which is asked.
The expected answer is: **3 1**



Questions

Write a program that detects the winning group of checkers on those boards:

1. 6x7 board : 6,4,4,4,1,3,6,1,6,6,3,5,1,3,7,4,4,2,6,2,1,4
2. 6x7 board : 6,3,2,3,5,3,4,5,1,6,4,5,3,3,2,3,4,7,1,1,5,2,2,2,6,4,6,4,4,1,6,6,5
3. 60x70 board : Connect4_q3.txt

15. Unpaid invoices (35pts +12/-2)

Introduction

Your father wants to make his accounts. Since he has no software to do it for him, he has to check each month by hand who the defaulters are. Fortunately, during the last family meal, you shiny Sopra-Sterien, told him it would not be complicated to automate the task, all you need are the amounts, dates and invoice numbers ... Finally having the chance to demonstrate you're not only a printer problems solver, you also surprised yourself saying "I can code this in less than half an hour, Dad, it's a piece of cake! ". So…, you have 30 minutes to show what you're made of! Who's the boooooosssssssssss????

Rules

- The input file is a **non ordered** list of credits and invoices formatted as A BBB CCC.CC DDDDDDDD
  - A is the character « I » (for Invoice) or « C » (for Credit)
  - BBB is the unique number of the invoice or credit.
  - CCC.CC is the amount of the invoice or credit.
  - DDDDDDDD is the date of the invoice or credit, formatted as YYYYMMDD
- The purpose of the exercise is to find out what are the non-credited invoice (i.e. unpaid)
- The credit is held in maximum 3 days after the invoice (inclusive)
- Some credits are not associated with any invoice. This should not interfere with treatment
- There cannot be any ambiguity. If there are two invoices of the same amount on the same day, there cannot be only one credit within 3 days. Either both invoices are paid, or both are unpaid. (see examples)
- A credit must not be associated to more than 1 invoice.
- The expected answer is the list of unpaid invoices, sorted in ascending order, separated by commas (for instance: **001,005,123**).

Examples :

- Simple :

Input:

C 001 100.00 20151202
I 002 110.00 20151205
C 003 110.00 20151213
I 004 100.00 20151201
I 005 110.00 20151210

004 is paid by 001, 005 is paid by 003.
Expected answer : **002**

- Complexe :

Input (sorted for better comprehension):

I 001 100.00 20151201
I 002 100.00 20151201
C 003 100.00 20151201
C 004 100.00 20151202
I 005 110.00 20151204
I 006 110.00 20151205
I 007 100.00 20151206
C 008 110.00 20151206
I 009 100.00 20151206
I 010 110.00 20151206

001 and 002 are paid by 003 and 004.
Invoices 005, 006, and 009 precede 008.
005 is paid by 008 since it is the earliest.
The expected answer is: **006,007,009,010**

- Unallowed case :

Input:

I 001 100.00 20151201
I 002 100.00 20151201
C 003 100.00 20151202
I 004 100.00 20151210

It is **not possible** to know whether invoice 001 or 002 has been paid by credit 003. This case will not be allowed in this exercise. Nevertheless, an additional credit like "C 005 100.00 20151204" would solve the problem and make the data-set valid (even if the credit 003 is not the same day!).

Questions :

1) I 001 123.45 20160111
I 002 123.45 20160102
I 003 123.45 20160106
C 004 123.45 20160104
C 005 123.45 20160114

2) C 001 103.42 20151207
C 002 432.27 20151205
C 003 845.36 20151202
I 004 845.36 20151201
I 005 432.27 20151206
C 006 845.36 20151201
I 007 502.91 20151206
I 008 432.27 20151204
C 009 432.27 20151206
I 010 845.36 20151201
I 011 432.27 20151201

3) File Invoices_q3.txt. Be careful, there are no more than 5 unpaid invoices

16. Building an automated Lego Pick-up (45pts +15/-2)

Introduction

Jack, computer and new technology enthusiast, seeks to create his own robot using Lego, and adding intelligent components. So he wishes to buy a Rasberry Pi, a camera, utlrasonic sensors, motors, memory cards ... all that stuff ... The problem is that, by talking about it, the challenge is launched among all his workmates. So they want to regroup their purchasing to save money! In addition, Peter already has some parts, Andrew also. Help them calculate the price of their order.

Rules

- Is provided as input the list of items they need
- Here are the different articles:
  - L = The Lego pick-up: 100€
  - M = Motors: 60€
  - C = Camera & Connectors: 50€
  - R = The Raspberry Pi 2: 40€
  - G = 16Go Memory class 10: 25€
- The website offers the following discounts:
  - **MULTIPACK** : For each LMCRG pack purchased, the next is 15% less than the previous one:
    - 1Pack = 275€
    - 2Packs = 275+233.75 = 508.75€
    - 3Packs = 275+233.75+198.69=707.44€
    - 4Packs = 275+233.75+198.69+168.89=876.33€
  - When calculating the MULTIPACK, each price is rounded to the cent (0.505 rounds to 0.51). The calculated price of the 4th package is 168.8865, rounded to 168.89. The next package will be 168.89 * 0.85 = 143.5565 ≈ 143.56 (not 168.8865 * 0.85 = 143.553525 ≈ 143.55).
  - **PACK DUO+**: If you buy 2 articles, the cheapest one has 50% off.
    - L+M = 130€
    - L+R = 120€
    - C+C = 75€
    - Etc...
  - For each article, a single reduction is applicable. MULTIPACK or PACK DUO+ or none.
  - Promotions can be mixed in the cart.
- The wanted price is the most favorable prices possible with 2 decimal places, and with a "."(dot) as separator. For example: **123.45**
- Question 3 contains a large data set.
  - You may try a brute force... But you may also have problems...
- It is not necessary to know the best method... you can compare several hypotheses...

Example

Input : LMCRGLLMM
Possible solutions :

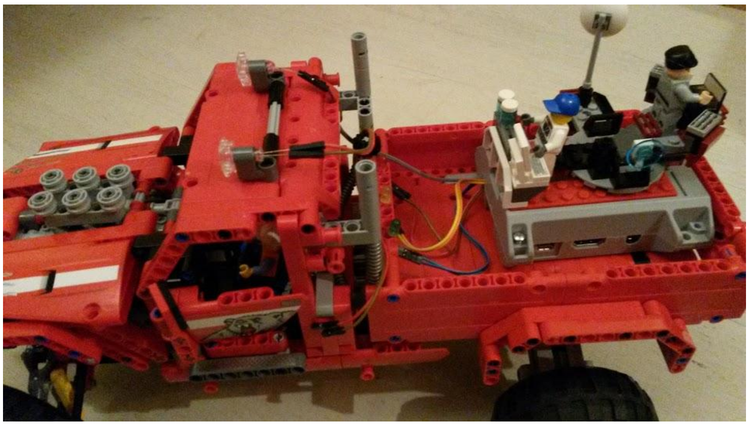
- MG + LC + MR + ML + L = 72.5+125+80+130+100=507.50€
- LMCRG + LM + LM = 275+130+130 = 535.00€
- LMCRG + LL + MM = 275+150+90 = 515.00€
- LM + CR + GL + LM + M = 130+70+112.50+130+60=502.50€
- LM + CR + G + LL + MM = 150+130+90+70+25=465.00€
- Etc...

Here, the best price is **465.00**

Questions

Write a program that calculates the cheapest cart with the following parts:

1. MLLMLLCLR
2. GRRRLRRGMLMRGGCGMGMLMRGCRCCRRLGRCMCGRCGRLCGLMMLRGLMMLCCGRGRMMGRRRCLCR
3. File Pickup_q3.txt



17. Battle code rocks language (50pts +18/-2)

Introduction

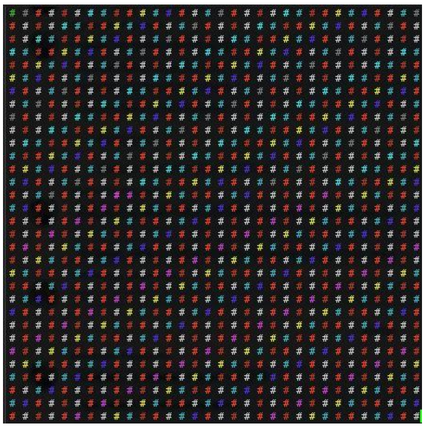
This language offers us to write a message using only 7 custom instructions.

Rules

This language works with an array of integers of indefinite size, with all values initialized to 0.
It uses a pointer referencing a table cell (initially the first).
There will be no mistakes in the provided code. Each “ro” is associated to a “cks”

The instructions are:

- **Ba** : moves the pointer to the next cell
- **tt** : if the pointed cell is not the first, moves the pointer to the previous cell
- **le** : increments the value of the pointed cell
- **co** : decrements the value of the pointed cell
- **de** : displays to standard output the character represented by the ASCII code defined by the value of the pointed cell
- **ro** : if the pointed cell’s value is 0, jump just after corresponding cks instruction
- **cks** : if the pointed cell’s value is not 0, return to the corresponding ro instruction



« ro/cks » loops can be nested:

- ro
--ro
---(instructions)
--cks
-cks

Questions

Write a program that provides the string generated by the following code:

1) lelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelelededecocode
2) leleleleleleleleleroBaleBaleleBaleleleBaleleleleBaleleleleleBaleleleleleleBaleleleleleleleBalelelelelelelele
BaleleleleleleleleleleBaleleleleleleleleleleleBaleleleleleleleleleleleleBaleleleleleleleleleleleleleBalelelelelelelelel
eleleleleBaleleleleleleleleleleleleleleleBaleleleleleleleleleleleleleleleleBalelelelelelelelelelelelelelelelelettttttttttt
ttttttttttttttttttttttcocksBaBaBaBaBaBaBaBaBaBaleleledecococoBaBaBaBacodeledeleledecocottttledecott
ttttttttttttttdeBaBaBaBaBaBaleleledecococoBaBaBaBaBacococococodelelelelettttcococodeleleleBaleledec
ocottledecottledecottttttttttttttttttttttttttde
3) File Battle_code_rocks_q3.txt