YEAR 12 COMPUTER SCIENCE PROGRAMMING CHALLENGES

1. Sign up for a GitHub account if you don't have one already - https://github.com/
   - Create a new repository with the following options:

     Name: Programming challenges
     Description: Miscellaneous programming challenges
     Public
     Add a README FILE
     .gitignore template: Python
     License: GNU General Public License v3.0

2. Make the following directory structures in your local area:
   - **N:/Github/Codewars**
   - **N:/Github/Brampton Manor Academy**

3. Attempt these programming challenges in any order
   - Write these programs using functions before attempting anything else
   - Keep organised, each challenge (and the PDF explaining it) should go in the BMA GitHub folder you made in step 2
   - Test your code! **testing.py** was given to you with this document, each challenge must have this in the same folder1

4. When a challenge is complete, upload your code to GitHub, maintaining the directory structure and provide the link to that specific challenge folder on Google Classroom

Y12CS@BMA

1 - ROD CONVERSIONS

Conversions are useful both in science and daily life. Here we examine some obscure, but useful conversions as well as some silly ones.

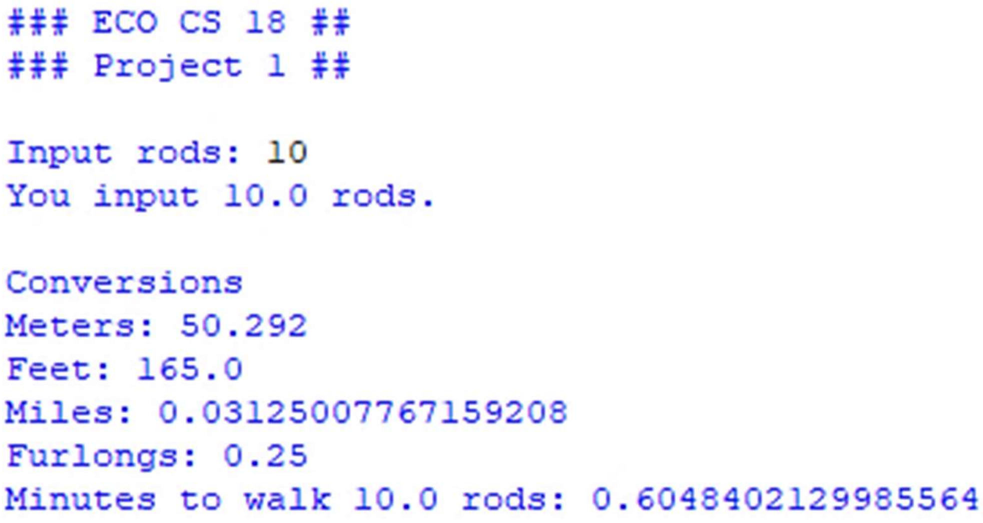
Your program will prompt the user for a floating-point value representing a distance in rods. You will reprint that value along with that value converted to the following values. The most important value is the time to walk the distance.

- Meters
- Feet
- Miles
- Furlongs
- The time in minutes to walk that distance

You can find these measures on the web, but so everyone is using the same conversions:

- 1 rod = 5.0292 meters
- 1 furlong = 40 rods
- 1 mile = 1609.34 meters
- 1 foot = 0.3048 meters
- average walking speed is 3.1 miles per hour

Sample output:



```
### ECO CS 18 ##
### Project 1 ##

Input rods: 10
You input 10.0 rods.

Conversions
Meters: 50.292
Feet: 165.0
Miles: 0.03125007767159208
Furlongs: 0.25
Minutes to walk 10.0 rods: 0.6048402129985564
```

2 - RICHTER

For each of the following Richter scale measurements, your program will perform the appropriate calculations and display the equivalent amount of energy in joules and in tons of exploded TNT:

- 1.0
- 5.0
- 9.1 (Indonesia earthquake, 2004)
- 9.2 (Alaska earthquake, 1964)
- 9.5 (Chile earthquake, 1960; largest ever measured)

Your program will then prompt the user to enter a Richter scale measurement, accept a floating point value representing that measurement, perform the appropriate calculations, and display the equivalent amount of energy in joules and in tons of exploded TNT for that user-selected value.

The Richter scale is a way to quantify the magnitude of an earthquake using a base-10 logarithmic scale. The magnitude is defined as the logarithm of the ratio of the amplitude of waves measured by a seismograph to an arbitrarily small amplitude.

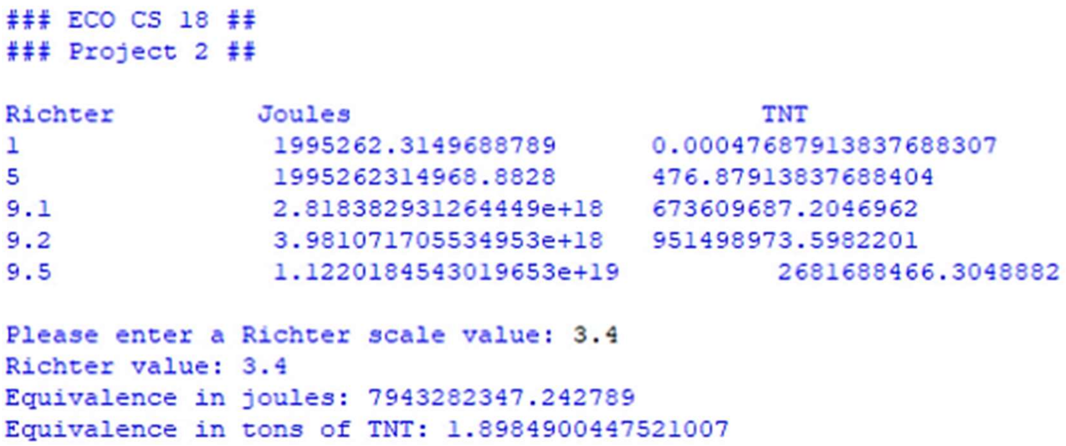
The energy in joules released for a particular Richter scale measurement is given by:

$$\text{Energy} = 10^{(1.5*\text{richter})+4.8}$$

where *Energy* is measured in *joules* and *richter* is the Richter scale measurement (typically on a scale from 1-10 as a floating point number).

One ton of exploded TNT yields 4.184x10^9 joules. Thus, you can relate the energy released in joules to tons of exploded TNT.

Sample output:



```
### ECO CS 18 ##
### Project 2 ##

Richter         Joules                          TNT
1                1995262.3149688789      0.00047687913837688307
5                199526231496 8.8828      476.87913837688404
9.1              2.818382931264449e+18   673609687.2046962
9.2              3.981071705534953e+18   951498973.5982201
9.5              1.1220184543019653e+19          2681688466.3048882

Please enter a Richter scale value: 3.4
Richter value: 3.4
Equivalence in joules: 7943282347.242789
Equivalence in tons of TNT: 1.8984900447521007
```

3 – THE 99 TRICK

There are two people in this game, you and a friend.

Your part:

- Select a number between 10 and 49. This is the **answer**.
- Calculate 99 – **answer** and remember it. This is **factor**.

Friends part:

- Have your friend select a number between 50 and 99.
- Add the **factor** from above to that number.
- Remove the hundred's digit and add it to the units digit.
- Subtract this number from your friend's original number.
- The result should be the **answer** from above.

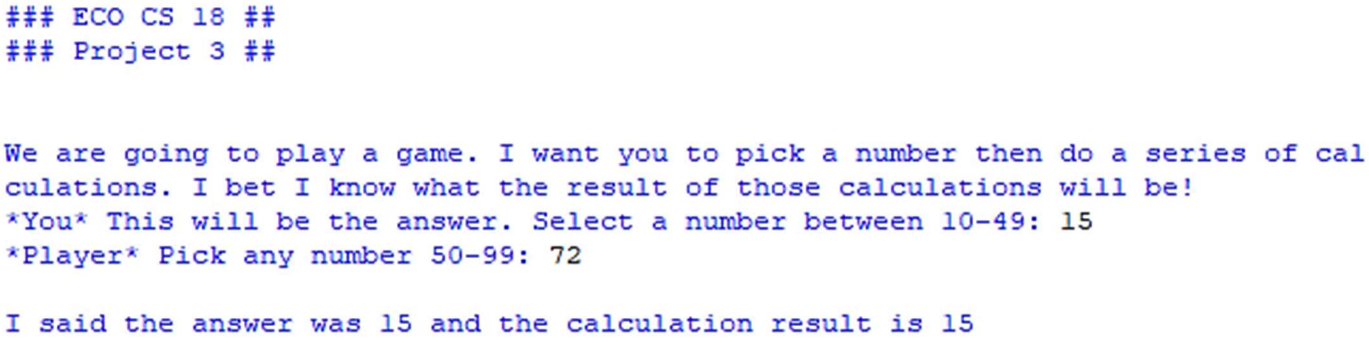
For example.

Your part:

- Pick 15 as the **answer**.
- Subtract 15 from 99, for a **factor** of 84.

Friend part:

- Pick a number: 72.
- Add 72 and 84 (the **factor**) for 156.
- Remove the hundred's digit and add it to the unit digit: 156 -> 56 + 1 -> 57.
- Subtract the picked number and the previous result, 72 – 57 giving 15, the **answer**.



```
### ECO CS 18 ##
### Project 3 ##


We are going to play a game. I want you to pick a number then do a series of cal
culations. I bet I know what the result of those calculations will be!
*You* This will be the answer. Select a number between 10-49: 15
*Player* Pick any number 50-99: 72

I said the answer was 15 and the calculation result is 15
```

4 – WIND CHILL

When air moves across exposed skin, humans perceive the temperature to be colder than the measured air temperature. There is a formula to quantify that perception which is frequently being updated to ensure that it is as accurate as possible.

For each of the following air temperature and wind speed measurements, your program will perform the appropriate calculations and display the corresponding wind chill temperature index:

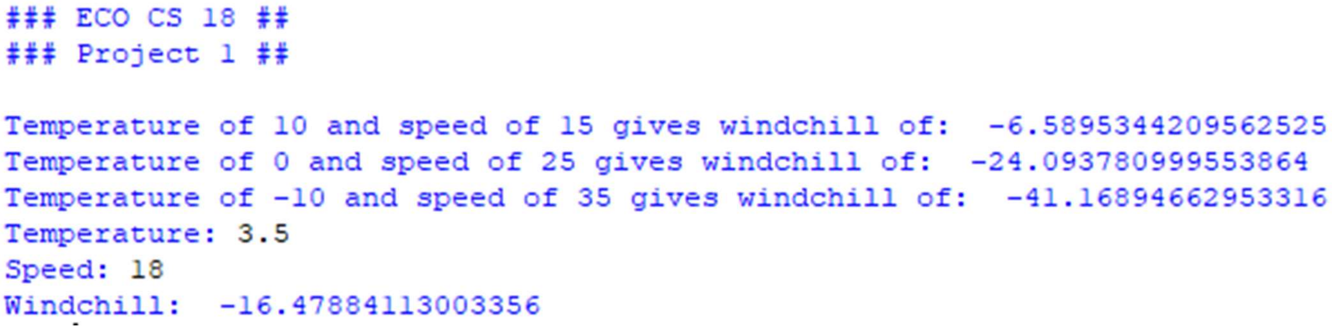
- 10.0 degrees and 15 MPH
- 0.0 degrees and 25 MPH
- -10.0 degrees and 35 MPH

After calculating and displaying the wind chill temperature indices for the three pairs of measurements given above, your program will then:

- prompt the user to enter an air temperature measurement and accept a floating point value representing that measurement
- prompt the user to enter a wind speed measurement and accept a floating point value representing that measurement
- perform the appropriate calculations
- display the wind chill temperature index for those user-selected values

The following Python statement can be used to calculate the Wind Chill Temperature Index, assuming that variables "air_temp" and "air_speed" have been assigned values:

35.74 + 0.6215 * air_temp - 35.75 * air_speed**0.16 + 0.4275 * air_temp * air_speed**0.16



```
### ECO CS 18 ##
### Project 1 ##

Temperature of 10 and speed of 15 gives windchill of:  -6.5895344209562525
Temperature of 0 and speed of 25 gives windchill of:  -24.093780999553864
Temperature of -10 and speed of 35 gives windchill of:  -41.16894662953316
Temperature: 3.5
Speed: 18
Windchill:  -16.47884113003356
```

5 – SLAYER

One strategy for finding a solution to a number puzzle is to describe the solution by a set of algebraic constraints and then solve the constraints. Consider, for instance, the following puzzle:

United and the City are the names of two rival dance teams. United say to City: If one of you joins our team, then our team will be double the size of yours. City reply: If one of you joins our team, then the sizes of our teams will be equal. What are the sizes of the two teams?

If the sizes of United and City are denoted by u and c, respectively, then any solution to this puzzle satisfies the constraints: $(u + 1) = 2(c – 1)$ and $c + 1 = u – 1$. Solving these constraints yields a solution to the puzzle, namely, $u = 7$ and $c = 5$.

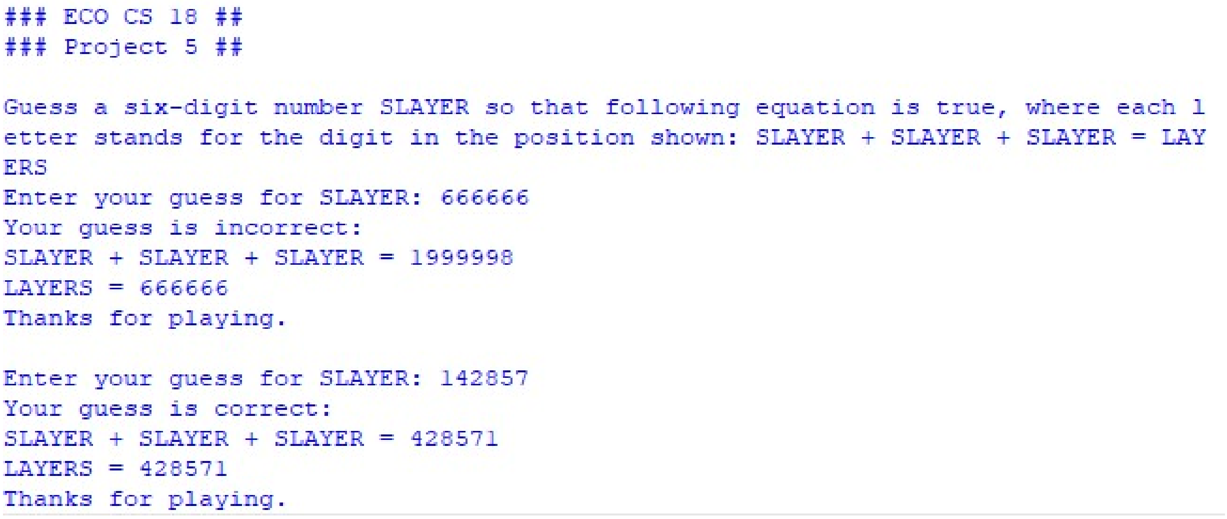
While solving the constraints in this example is easy, solving constraints describing some number puzzles can be too hard. In such cases, another strategy is to guess possible solutions and then check if the guess works. Consider, for instance, the following puzzle:

What digits can replace the letters A, B and C to make a 3-digit number ABC for which the following equation is true: $ABC = A \cdot B \cdot C \cdot (A+B+C)$

This puzzle is not easily solved directly. But any guess for the 3-digit number ABC can be easily checked to see if it satisfies the equation. For example, suppose you guess that ABC is 123. This guess does not work because it requires A = 1, B = 2, and C = 3, and $123 \neq 1 \cdot 2 \cdot 3 \cdot (1+2+3)$. In contrast, the guess that ABC is 135 works since $135 = 1 \cdot 3 \cdot 5 \cdot (1+3+5)$.

You will write a program that can be used to check guesses for the following puzzle:

For what six-digit number SLAYER is the following equation true, where each letter stands for the digit in the position shown: SLAYER + SLAYER + SLAYER = LAYERS



```
### ECO CS 18 ##
### Project 5 ##

Guess a six-digit number SLAYER so that following equation is true, where each l
etter stands for the digit in the position shown: SLAYER + SLAYER + SLAYER = LAY
ERS
Enter your guess for SLAYER: 666666
Your guess is incorrect:
SLAYER + SLAYER + SLAYER = 1999998
LAYERS = 666666
Thanks for playing.

Enter your guess for SLAYER: 142857
Your guess is correct:
SLAYER + SLAYER + SLAYER = 428571
LAYERS = 428571
Thanks for playing.
```

6 – LATIN SQUARES

A Latin Square is an n*n table filled with n different symbols in such a way that each symbol occurs exactly once in each row and exactly once in each column (see http://en.wikipedia.org/wiki/Latin_square).

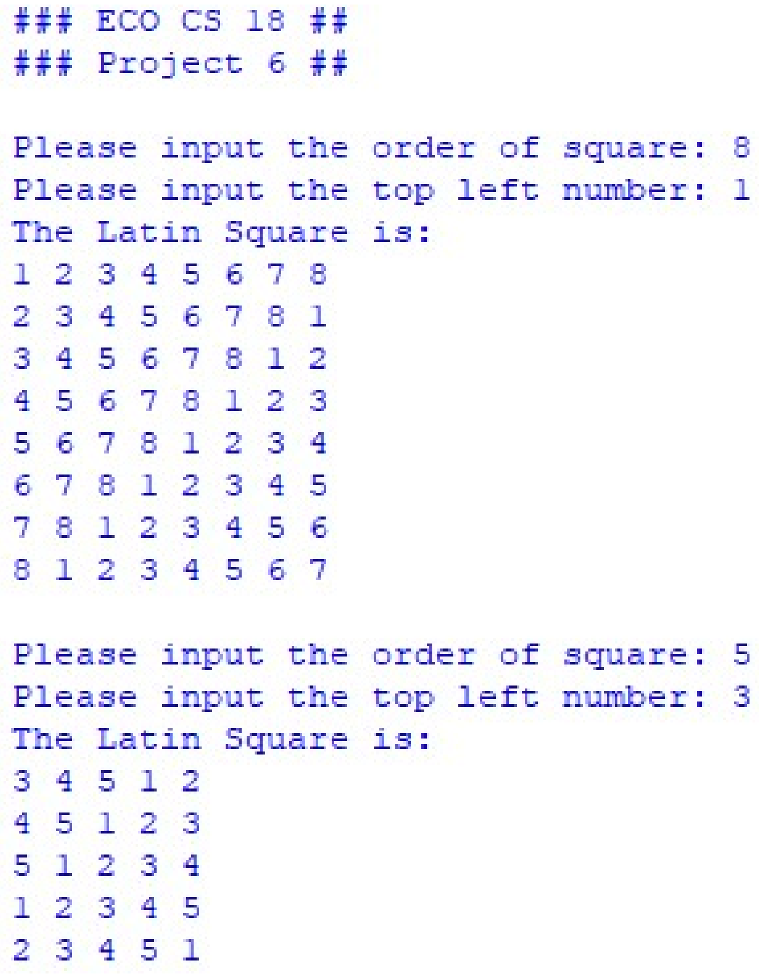
For example, two possible Latin Squares of order 6:

1 2 3 4 5 6
2 3 4 5 6 1
3 4 5 6 1 2
4 5 6 1 2 3
5 6 1 2 3 4
6 1 2 3 4 5
	3 4 5 6 1 2
	4 5 6 1 2 3
	5 6 1 2 3 4
	6 1 2 3 4 5
	1 2 3 4 5 6
	2 3 4 5 6 1

Obviously, the top-left numbers are 1 and 3 respectively.

Your program will ask user to input two numbers. The first number is the order of square; the second one is the top-left number of the square. Note that the second number should be between 1 and the first number, so your program should check this situation.

Then, your program will print the corresponding Latin Square.



```
### ECO CS 18 ##
### Project 6 ##

Please input the order of square: 8
Please input the top left number: 1
The Latin Square is:
1 2 3 4 5 6 7 8
2 3 4 5 6 7 8 1
3 4 5 6 7 8 1 2
4 5 6 7 8 1 2 3
5 6 7 8 1 2 3 4
6 7 8 1 2 3 4 5
7 8 1 2 3 4 5 6
8 1 2 3 4 5 6 7

Please input the order of square: 5
Please input the top left number: 3
The Latin Square is:
3 4 5 1 2
4 5 1 2 3
5 1 2 3 4
1 2 3 4 5
2 3 4 5 1
```

7 – 196 ALGORITHM

The 196-algorithm is a procedure for creating a palindromic integer: an integer that has the same value when examined forwards or backwards. Examples of palindromic integers are: 88, 121, 2332, 12321, etc. The 196-algorithm is as follows.

1. If the integers is a palindrome, then print that integer
2. Otherwise, take the integer and its reversal and add them together.
3. With the sum, repeat the process starting at step 1.

Here are some examples:

56: 56+65=121! palindrome

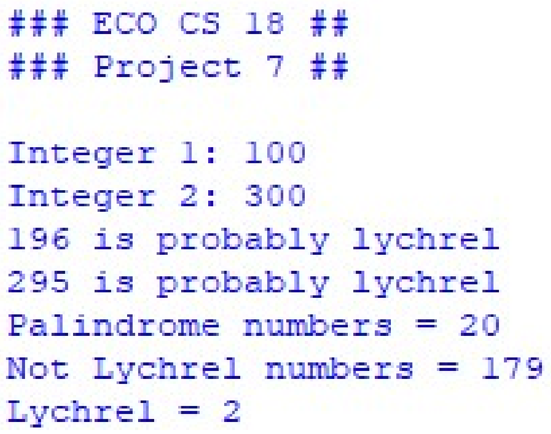
57: 57+75=132; 132+231=363! palindrome

87: 87+78=165; 165+561=726; 726+627=1353; 1353+3531=4884! palindrome

89: 24 steps to the palindromic number 8813200023188

It is called the 196-algorithm because the integer 196 is the first number that, it appears, does not converge to a palindromic number. Such a number is called a Lychrel number (see http://goo.gl/p3ATWp). Though it hasn't been mathematically proven that 196 doesn't converge, it has been shown to not converge out to a number containing 300 million digits!

1. Prompt for two integers. These two integers constitute the range of integers (inclusive) of the integers that will be checked.
2. After the program runs, you will report the following statistics for the numbers examined in the given range:
   a. The number of "natural" palindromes (numbers in the range that are already palindromes, such a 11, 121, 3553)
   b. The number of non-Lychrel numbers (numbers which eventually yield a palindrome using the 196 algorithm)
   c. The number of Lychrel numbers encountered. Assume a maximum of 60 iterations to indicate a Lychrel number
   d. Because Lychrel numbers are rare, report each Lychrel number as it occurs.



```
### ECO CS 18 ##
### Project 7 ##

Integer 1: 100
Integer 2: 300
196 is probably lychrel
295 is probably lychrel
Palindrome numbers = 20
Not Lychrel numbers = 179
Lychrel = 2
```

8 – THROBAC

In 1937 Claude Shannon showed how Boolean logic could be expressed in electronic circuits leading to computers as we know them. For fun he created Throbac (THrifty ROman-numerical BAckward-looking Computer), a calculator that did arithmetic with Roman numerals (pictured above)—a useless, but whimsical device. In this project, you will write a program that performs addition with Roman numerals, just like Throbac.

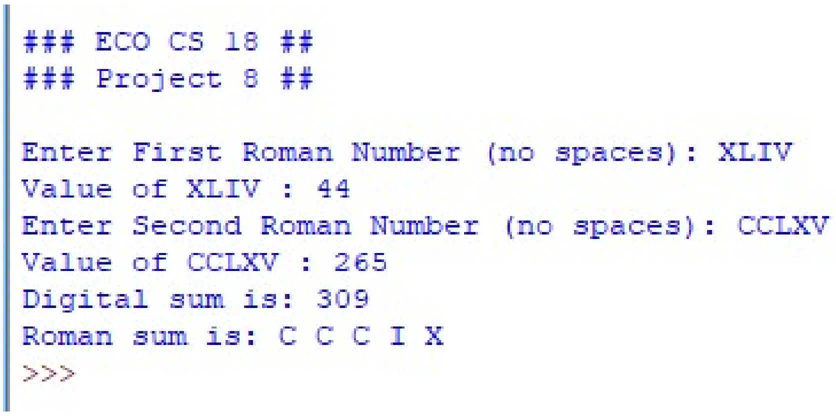
Roman numerals have values, but are not positional like our number system, and there is no zero (zero arrived in the west circa 1000 AD). To keep things simpler, we will use five of the seven symbols: I = 1, V = 5, X = 10, L = 50, and C = 100.

Numbers are formed by combining symbols together and adding their values—only whole numbers. For example, CLXVII is 100+50+10+5+1+1 = 167. Generally, symbols are placed in order of value, starting with the largest values. When smaller values precede larger values, the smaller values are subtracted from the larger values, and the result is added to the total. For example, in XLIV the smaller X before L means subtract X from L as 50-10 to get 40, and the I before V means subtract I from V as 5-1 to get 4, so the final number is 44. Some call that the "subtraction rule." A useful restriction of the subtraction rule is that there can never be more than one smaller value. That is, XL is valid, but XXL is not: thirty is represented by XXX.

Hint: Solve the problem for Roman Numerals that don't use the subtraction rule, e.g. assume input is something like LXXII and not numbers like XLIV. After you get that working, introduce the subtraction rule. Assume that the values will be less than 380.

Your program will:

1. Prompt for two Roman Numerals.
2. Convert both numbers to integers.
3. Find the sum of the two integers.
4. Print the sum as a Roman Numeral.



```
### ECO CS 18 ##
### Project 8 ##

Enter First Roman Number (no spaces): XLIV
Value of XLIV : 44
Enter Second Roman Number (no spaces): CCLXV
Value of CCLXV : 265
Digital sum is: 309
Roman sum is: C C C I X
>>>
```

9 - TURTLE ANGLE CALC

Your program will read in 2 points from the user, and then use the methods in the turtle module to draw a line from 0,0 to the 1st point, and from the 1st point to the second point. You will also use geometry to calculate the acute (smaller) angle between the 2 lines you draw.

Main components of program

1. output a brief, descriptive message when it first starts, indicating the purpose of the program and the user-required input and output that will be provided.
2. prompt the user for the x coordinate of the first point, then for the y coordinate. Repeat for the 2nd point.
3. print the acute angle between the 2 lines (1st line from 0,0 to 1st point, 2nd line from 1st point to 2nd point).
4. using the turtle module, draw the 2 lines on the screen. At the end of the 2nd line, write the angle previously calculated.

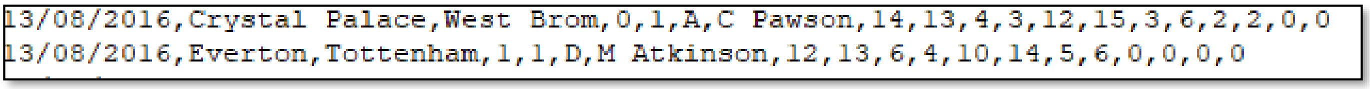
Useful information

- To use turtle, it must be imported: **import turtle**
- The turtle will always start at (0,0)
- To make the turtle move, for now use **turtle.goto(x,y)**
- To write using the on the screen with the turtle use **turtle.write(x)**
- To finish using the turtle use **turtle.done()**
- To calculate the acute angle between two lines where (x1, y1) and (x2, y2) are 2 points on the line the formula is: $\frac{y_2 - y_1}{x_2 - x_1}$
- To calculate the angle between two lines where m1 and m2 are the slopes of the 2 lines, and m2 is larger than m1 the formula is: $\tan\phi = \frac{m_2 - m_1}{1 + m_1 m_2}$
- To ensure that you always subtract the smaller value from the larger, you will need the absolute value which is built into python with: **abs()** e.g. abs(-47) = 47
- To use tan, you must import math: **import math**
- The tan function **atan** returns radians, to convert to degrees the formula is **(angle in radians)*180/pi**
- Pi can be accessed with **math.pi**

10 – LEAGUE TABLE

Quite often as a programmer you are given a set of data that requires processing. Although the subject of the data may not interest you, what is important is that the outcome is accurate.

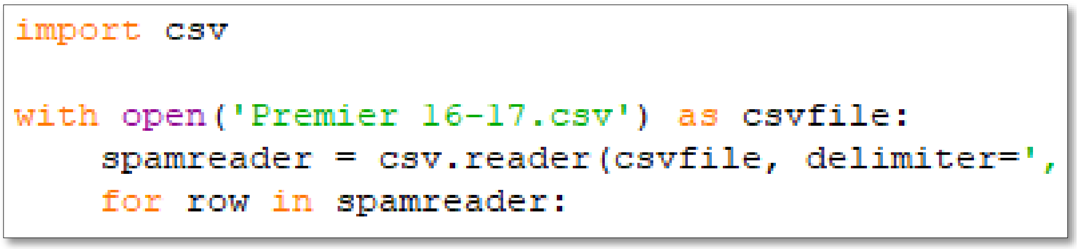
In this challenge you have been provided with a csv file containing some of the data generated from the 16-17 premier league football season. CSV stands for Comma Separated Value, an example of which is below:



```
13/08/2016,Crystal Palace,West Brom,0,1,A,C Pawson,14,13,4,3,12,15,3,6,2,2,0,0
13/08/2016,Everton,Tottenham,1,1,D,M Atkinson,12,13,6,4,10,14,5,6,0,0,0,0
```

Programs can read these files easily into lists or arrays because the comma separates each value into an index. As an example, if each line was saved as a list called row then row[3] would be '0' on the first line and '1' on the second line.

To access the csv file and loop through it row by row, the following code could be used:



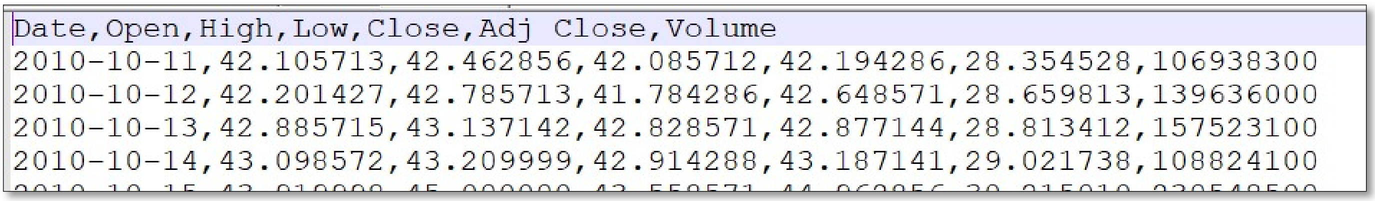
```
import csv

with open('Premier 16-17.csv') as csvfile:
    spamreader = csv.reader(csvfile, delimiter=',
    for row in spamreader:
```

The main task of this challenge is to process the data file and produce the following pieces of information:

1. A league table showing: wins, draws, losses, goal difference and total points. Team with most points at the top.
    a. Wins are three points for the winning
    b. Draw is one point for both teams
    c. Loss is 0 points for the losing team
    d. Ties in total points use goal difference to separate (scored – conceded)
2. Single statistics for the following information points:
    a. Most accurate team (Total shots on target / total shots)
    b. Least accurate team (Total shots on target / total shots)
    c. Dirtiest team (most fouls per game)
    d. Cleanest team (least fouls per game)
    e. Referee with highest card average per game (Yellows +1, Red+2)
    f. Referee with lowest card average per game (Yellows +1, Red+2)

11 – STOCK DATA MINING

Data mining is the process of sorting through large amounts of data and picking out relevant information. In this project you will do an analysis of Apple stock prices. Your program will calculate the monthly average prices of Apple stock from 2010 to 2018 and say the best 6 months and the worst 6 months. The CSV file is as follows:



```
Date,Open,High,Low,Close,Adj Close,Volume
2010-10-11,42.105713,42.462856,42.085712,42.194286,28.354528,106938300
2010-10-12,42.201427,42.785713,41.784286,42.648571,28.659813,139636000
2010-10-13,42.885715,43.137142,42.828571,42.877144,28.813412,157523100
2010-10-14,43.098572,43.209999,42.914288,43.187141,29.021738,108824100
```

How to calculate the average price?

Total sales for a day = Adj Close * Volume

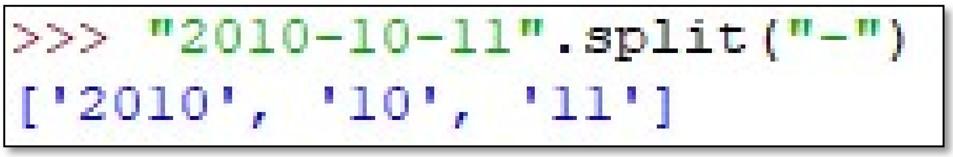
The average price for two days=

(Adj Close$_1$ * Volume$_1$ + Adj Close$_2$ * Volume$_2$) / (Volume$_1$ + Volume$_2$)

To average a whole month, you just add up the total sales (V*C) for each day and divide by the sum of all the volumes (V$_1$ + V$_2$ + … + V$_n$).

How can I get the year and month?

The split method will allow you to separate a string by a character. As an example:



```
>>> "2010-10-11".split("-")
['2010', '10', '11']
```

This array can then be indexed using 0 for the year, 1 for the month and 2 for the day.

Decomposing the problem

1. Read the CSV file and loop through each row
2. Split the date into separate parts
3. Check whether you have stored that year and month before, if not create a storage container for it with a 3 index list e.g. [0,0,0]
4. Calculate **Adj Close * Volume** and then add it to **index [0]**
5. Add **Volume** to the previous values stored in **index [1]**
6. Calculate **index [0] / index[1]** and store the result in **index[2]**
7. Find a way to sort the lists and show the top 6 and bottom 6 months.

12 – CARD CHECKSUM

When you receive a card from the bank, various rules are followed to ensure that the 16-digit number across the front is valid. In this challenge you will follow the rules and determine where the card comes from and whether the number itself is valid.

Task 1: Is the number valid?

Check the card number against the following rules:

Rule	If error
Must be 16 digits	Print 'too short' or 'too long', ask again
Must have no letters	Print 'Card number must be just numbers', ask again

Task 2: Identify the Personal Account Number (PAN)

Digit 7 to 15 are the PAN.

Task 3: Separate the checksum digit

The last digit is the checksum

Task 4: Who issued the card?

Use the following table of rules to determine the issuer:

Leading digits	Issuer
34 or 37	American Express
Others beginning with 3	JCB
4	Visa
51 to 55	MasterCard

Task 5: Is the card number real?

To verify that a card is real we use the Luhn algorithm (https://en.wikipedia.org/wiki/Luhn_algorithm).

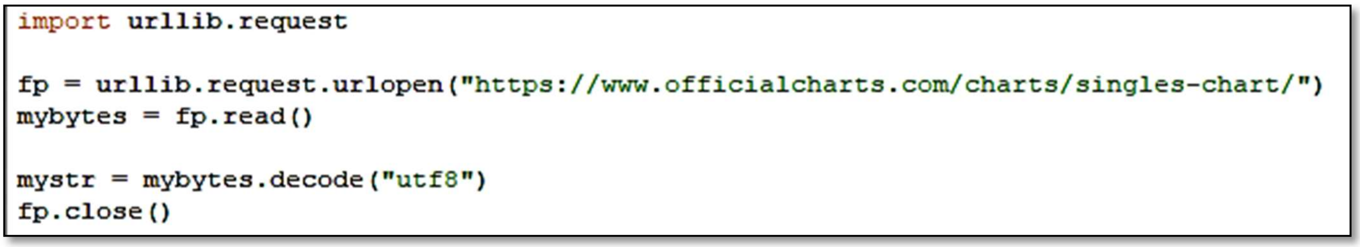
Valid cards to use: 4111111111111111, 5555555555554444

13 – SCRAPING SINGLES

Websites can have a lot of useful information on them. Sometimes though our program might need to get this information live. The only issue with doing this is that python will download the whole thing as a HTML string that would then need to be filtered through. Luckily for you, this is the task.

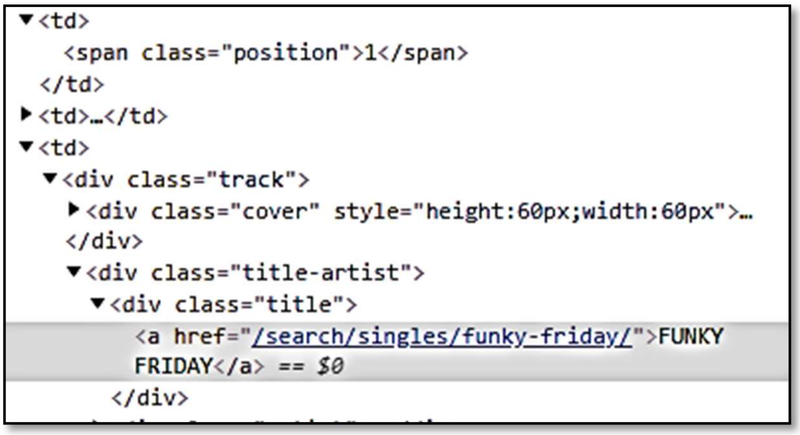
Step 1: Read the website into python

This code is pretty standard. The only thing different from the manual is the link that is being read. In this case, the link is: https://www.officialcharts.com/charts/singles-chart/



```
import urllib.request

fp = urllib.request.urlopen("https://www.officialcharts.com/charts/singles-chart/")
mybytes = fp.read()

mystr = mybytes.decode("utf8")
fp.close()
```

Step 2: Understand the HTML



```
▼<td>
   <span class="position">1</span>
 </td>
▶<td>…</td>
▼<td>
  ▼<div class="track">
    ▶<div class="cover" style="height:60px;width:60px">…
    </div>
    ▼<div class="title-artist">
      ▼<div class="title">
          <a href="/search/singles/funky-friday/">FUNKY
          FRIDAY</a> == $0
        </div>
```

When you right click on a page and press inspect, you can view the code that created the page. Using this information, you can look for patterns that can be taken advantage of. This webpage has the following two tags that I am interested in:

<span class="position"> and **<a href="**. Each of these would allow me to splice the string down so that it just shows the position of the track in the chart and then the name of the track.

Step 3: Create the main loop

Ultimately what you have stored in the variable **mystr** is a massive string. This can be spliced using **mystr**[:]. How do we know where to splice it though? To search through **mystr** you would write **a = mystr.find("search here")** which would either return the index where the string "search here" started or -1 if not found. You could even write another search that starts after where a was found with this **b = mystr.find("search here", a+1).** So back to the main loop, well **while a > -1:** would be enough to get you started. Remember to do another check later to keep a changing e.g. **a = mystr.find("search here", a+1)**.

14 - CALCULATOR

Your calculator will perform addition, subtraction, multiplication, float division, integer division, modulo and exponentiation. It should accept floating-point values, and it will check for division by zero when a division of any type is selected.

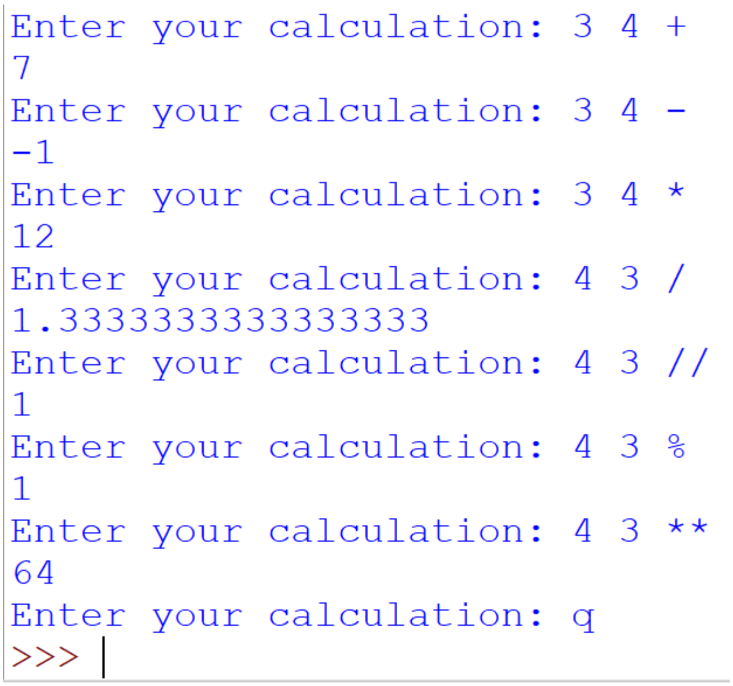
When the user enters the calculation, they must do it using Reverse Polish notation (RPN), where the operator follows the operand e.g. 3 4 + = 7.

Tasks

1. Ask the user for their calculation (Enter q for quit)
2. There must be three parts to the calculation, no more or less

   operand, operand, operator
3. The third part of the calculation must be one of the following operators

   + - * / // % **
4. Output the answer
5. Ask for the next calculation



```
Enter your calculation: 3 4 +
7
Enter your calculation: 3 4 -
-1
Enter your calculation: 3 4 *
12
Enter your calculation: 4 3 /
1.3333333333333333
Enter your calculation: 4 3 //
1
Enter your calculation: 4 3 %
1
Enter your calculation: 4 3 **
64
Enter your calculation: q
>>> |
```

15 – FRIENDLY FACES

The majority of websites are dynamically generated. Content is provided well in advance and then is manipulated without user interaction later. This allows sites to stay on top of trends and keeping up appearances.

You have been given a blank template for a website. The website needs content to make it useable, however the content has been given as a CSV file. On the page there are 15 placeholders that need to be swapped for the actual content. Each placeholder is numbered and named as follows: link1…link5, initials1…initials5 and name1…name5

Step 1: Read the CSV into python

Hint: You have this code already, it's also in the manual, you'll need to import something

Step 2: Read the HTML file into python

Hint: No need to import anything special for this, just open the file and read it

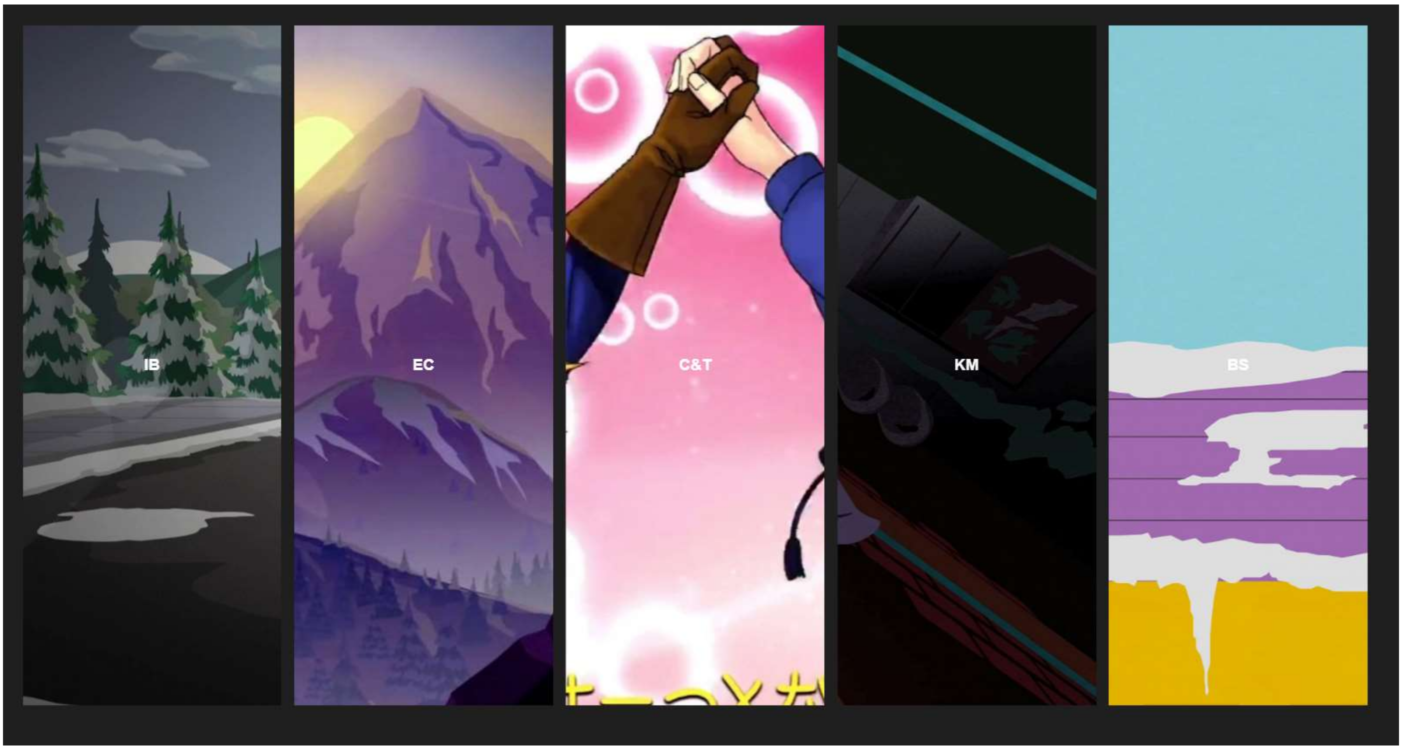
Step 3: Find and replace the placeholders with the information from the CSV

Hint: There are many ways of doing this: 5 loops, 15 replaces, find method…

Step 4: Save the HTML file back to the folder

Hint: Similar to the read method, except with write

Glorious output:



16 - DEBT MANAGEMENT

At the end of every month interest is added to a debt (initially 100) and then repayments are taken off. The interest is a fixed percentage of the current debt. The repayment is also a fixed percentage of the debt (after the interest has been added) or 50, whichever is larger. If the repayment is greater than the debt it is reduced to match the debt. The cycle of interest and repayment is continued until the debt has been reduced to 0 and paid off.

For example, suppose the interest is 10% and repayments are 50%:

- At the end of the first month the interest is 10, increasing the debt to 110;
- The repayment is 55 (50% of 110) leaving the debt at 55;
- At the end of the second month the interest is 5.5, increasing the debt to 60.5;
- The repayment is 50 (as 50% of 60.5 is less than the minimum amount), leaving the debt at 10.5;
- At the end of the third month the interest is 1.05, increasing the debt to 11.55;
- The repayment is 11.55 (the minimum payment of 50 is reduced to match the debt) and the debt is paid off.
- The total amount repaid on the debt is 116.55

When calculating percentages the amount is rounded up to 2 decimal places. E.g. 10.201 and 10.207 would both be rounded up to 10.21.

Task 1

Write a program that reads in the interest percentage followed by the repayment percentage. Percentages will be integers between 0 and 100 (inclusive).

You should output the total amount repaid. You will always be given input that allows the debt to be paid off.

Example: interest = 10, repayment = 50, output = 116.56

Task 2

Given an interest percentage of 43% and a repayment percentage of 46%, how many payments will be made to pay off the debt?

Change your code so that it calculates how many payments will be made.

17 - DECODER RING

A decoder ring consists of two adjacent dials and is used to encrypt (or decrypt) messages. The two dials are lined up, so that each position on the first is touching one on the second, and dials can rotate so that they can be aligned in different ways. The first dial has 26 positions, lettered from A to Z in order, and the second dial has the same letters but not necessarily in the same order. A letter is encrypted by finding it on the first dial and using the corresponding touching letter on the second dial.

For example, suppose that the second dial has been lettered from Z to A (i.e. reverse order) and that the A on the first dial is touching Z on the second:

- The letter A would be encrypted to the letter Z, the letter B by the letter Y etc.
- If the second dial is now rotated until the A on the first dial is touching X on the second, the letter A would be encrypted to the letter X, the letter B by W etc.

The order of the letters on the second dial will be generated as follows, from the number n:

- Place the letters A to Z clockwise in order around a circle;
- Starting with A count clockwise round the circle until n is reached;
- Remove the selected letter from the circle — it becomes the first letter on the second dial;
- Starting from where you left off, count to n again to select the next letter;
- Continue until all the letters have been selected.

For example, if n is 5 the letters will be chosen in the order: E, J, O, T, Y, D, K, Q, W…

The dials will be aligned so that the first letter selected for the second dial is initially touching the letter A on the first dial, the second letter selected touching B etc. After each rotation of the dials, the letter on the second dial previously touching B on the first dial will be touching A on the first dial, etc.

A word is encoded by encrypting each letter in turn, after each encryption rotating the dial by a single position. For example, if n is 5 the word ABCD will be encrypted as EOYK.

Task

Write a program that encrypts a word. The program should take a single integer (1 ≤ n ≤ 1000000) followed by an uppercase word (length of word: 1 ≤ w ≤ 8)

Example

Input: 5 ABCD **Output:** EOYK

18 - INCREMENTING PW

A password scheme accepts passwords that contain combinations of upper-case letters and digits. The scheme restricts passwords to ones in which letters appear in alphabetical order. The digits 0, ..., 9 are treated as coming after Z in alphabetical order; lower digits are treated as coming before higher digits alphabetically. Passwords must contain at least one character (letter or digit) and no duplicate characters are allowed.

For example:

- BIO14 is a valid password;
- OLYMPIAD is not a valid password (letters are not in alphabetical order).

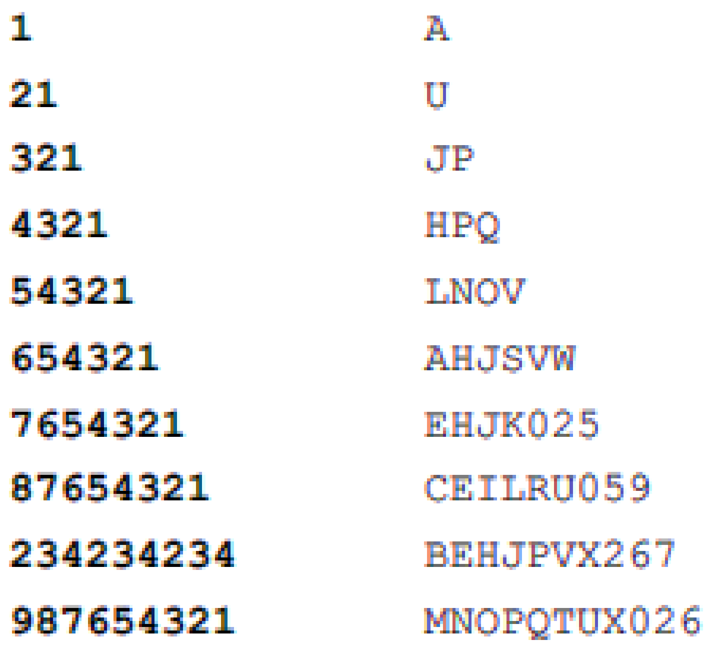
The passwords have been put into an ordered list. If two passwords contain a different number of characters, the password with the fewer characters comes first. If they both have the same number of characters then they are sorted alphabetically.

The ordered list of passwords looks like this: A, B, ..., Z, 0, 1, ... 9, AB, AC, ..., A9, BC, ...

Task

Write a program to determine the n^th^ password in the ordered list. Your program should read in a single integer, (1 ≤ n ≤ 1,000,000,000). You should output the string which represents the n^th^ password.

Test cases



1	A
21	U
321	JP
4321	HPQ
54321	LNOV
654321	AHJSVW
7654321	EHJK025
87654321	CEILRU059
234234234	BEHJPVX267
987654321	MNOPQTUX026

19 – MODERN ART

A gallery is displaying pieces of modern art by several artists and is considering the different ways of arranging the exhibition. As this is modern art all pieces by the same artist are indistinguishable.

For example, suppose there is a single piece by artist A, two by artist B and one by artist C. There are 12 ways the gallery might arrange the exhibition:

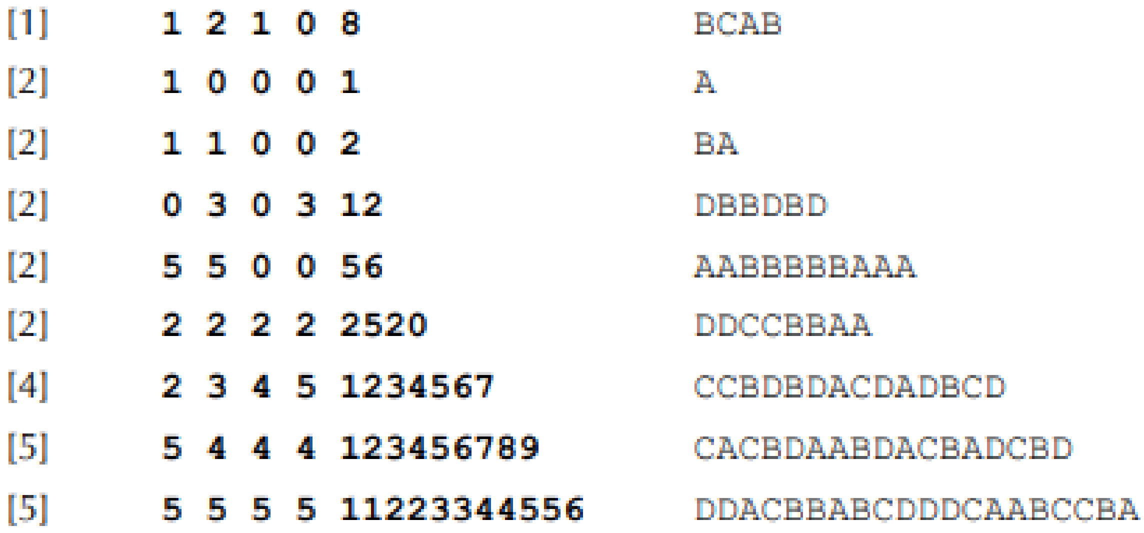
ABBC
ABCB
ACBB
BABC
BACB
BBAC
BBCA
BCAB
BCBA
CABB
CBAB
CBBA
ABBC

These have been listed in alphabetical order.

Write a program to determine the nth way of arranging the exhibition. Your program should input five integers: **a, b, c** and **d** (each between 0 and 5 inclusive) indicating the number of works by artists A, B, C and D in order, and finally **n**.

You will only be given input where at least one artist is exhibiting a work and n is no greater than the number of possible exhibitions.

You should output the string which represents the nth arrangement.



[1]	1 2 1 0 8	BCAB
[2]	1 0 0 0 1	A
[2]	1 1 0 0 2	BA
[2]	0 3 0 3 12	DBBDBD
[2]	5 5 0 0 56	AABBBBBAAA
[2]	2 2 2 2 2520	DDCCBBAA
[4]	2 3 4 5 1234567	CCBDBDACDADBCD
[5]	5 4 4 4 123456789	CACBDAABDACBADCBD
[5]	5 5 5 5 11223344556	DDACBBABCDDDCAABCCBA

20 - ?I?S?B?N?

An ISBN (International Standard Book Number) is a ten-digit code which uniquely identifies a book. The first nine digits represent the book and the last digit is used to make sure the ISBN is correct. To verify an ISBN you calculate 10 times the first digit, plus 9 times the second digit, plus 8 times the third...all the way until you add 1 times the last digit. If the final number leaves no remainder when divided by 11 the code is a valid ISBN.

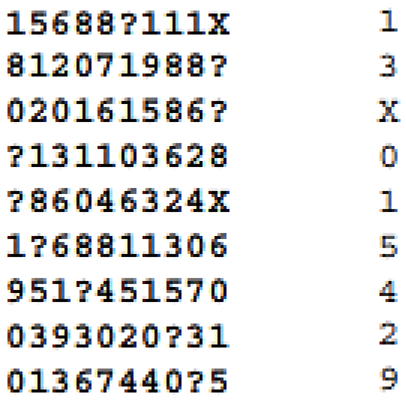
For example, 0201103311 is a valid ISBN, since 10*0 + 9*2 + 8*0 + 7*1 + 6*1 + 5*0 + 4*3 + 3*3 + 2*1 + 1*1 = 55.

Each of the first nine digits can take a value between 0 and 9. Sometimes it is necessary to make the last digit equal to ten; this is done by writing the last digit as X. For example, 156881111X.

Task

Write a program that reads in a valid ISBN with a single missing digit, marked with a ?, and outputs the correct value for the missing digit.

Test cases



15688?111X	1
812071988?	3
020161586?	X
?131103628	0
?86046324X	1
1?68811306	5
951?451570	4
0393020?31	2
0136744?75	9