01 Write a program that uses a dictionary that contains ten usernames and passwords. The program should ask the user to enter their username and password. If the username is not in the dictionary, the program should indicate that the person is not a valid user of the system. If the username is in the dictionary, but the user does not enter the right password, the program should say that the password is invalid. If the password is correct, then the program should tell the user that they are now logged in to the system.

02 Create a 5×5 list of random numbers between 1 and 20. Then write a program that creates a dictionary whose keys are the numbers and whose values are the how many times the number occurs. Then print the three most common numbers.

03 A DNA sequence is made by a combination of ATCG block combinations. Generate 20 random DNA sequences containing 8 blocks each. Write a code to match DNA sequence. User enters a set of blocks of a sequence and tries to find the possible match. An example is A**A****. The user would like to know which of the DNA in the list fit with their pattern.
In the example just given, the matching strings are the first and fourth. One way to solve this problem is to create a dictionary whose keys are the indices in the user's string of the non-asterisk characters and whose values are those characters. Write a program implementing this approach (or some other approach) to find the strings that match a user-entered string.

04 Python program to find the sum of all items in a dictionary

05 Python program to sort a dictionary based on its keys

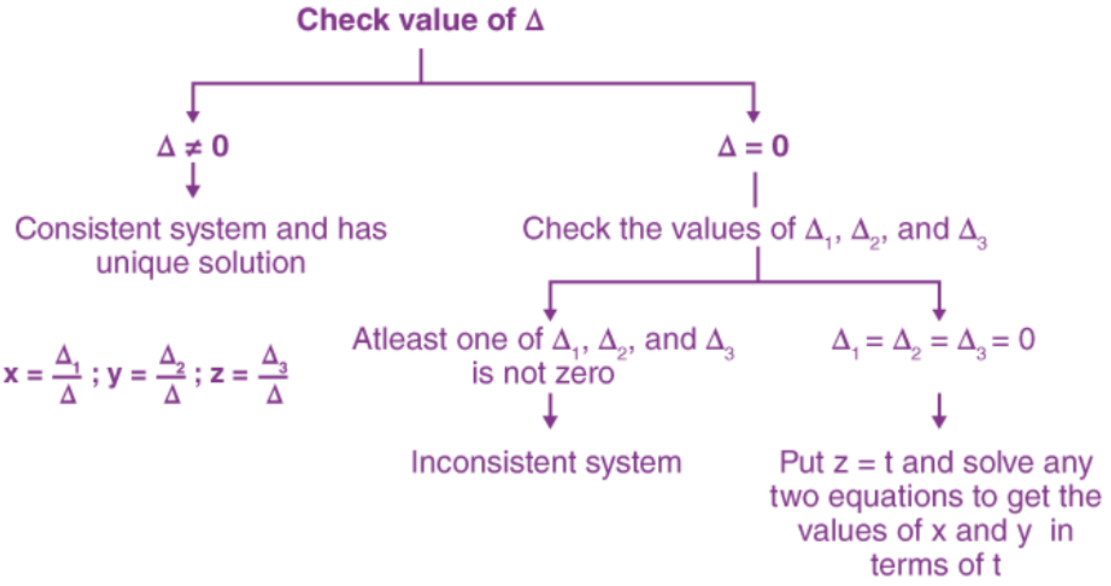
06 Python program to sort a dictionary based on its values

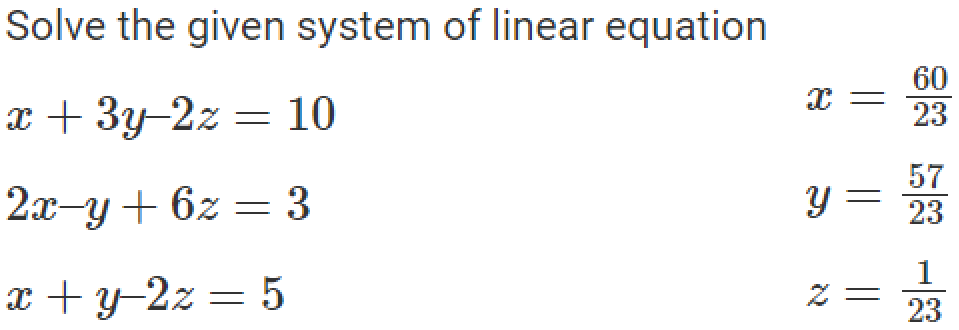
07 Write a Python program which solves simultaneous equation in three variables x, y, z (Consistent System)

$$a_1x + b_1y + c_1z = d_1 \;\; and \;\; a_2x + b_2y + c_2z = d_2 \;\; and \; a_3x + b_3y + c_3z = d_3$$

To solve this system we first define the following determinants

$$\Delta = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}, \Delta_1 = \begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix}, \Delta_2 = \begin{vmatrix} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{vmatrix}, \Delta_3 = \begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix}$$


Check value of Δ
Δ ≠ 0
Δ = 0
Consistent system and has unique solution
Check the values of Δ1, Δ2, and Δ3
x = Δ1/Δ ; y = Δ2/Δ ; z = Δ3/Δ
Atleast one of Δ1, Δ2, and Δ3 is not zero
Δ1 = Δ2 = Δ3 = 0
Inconsistent system
Put z = t and solve any two equations to get the values of x and y in terms of t



Solve the given system of linear equation

$$x + 3y - 2z = 10$$

$$2x - y + 6z = 3$$

$$x + y - 2z = 5$$

$$x = \frac{60}{23}$$

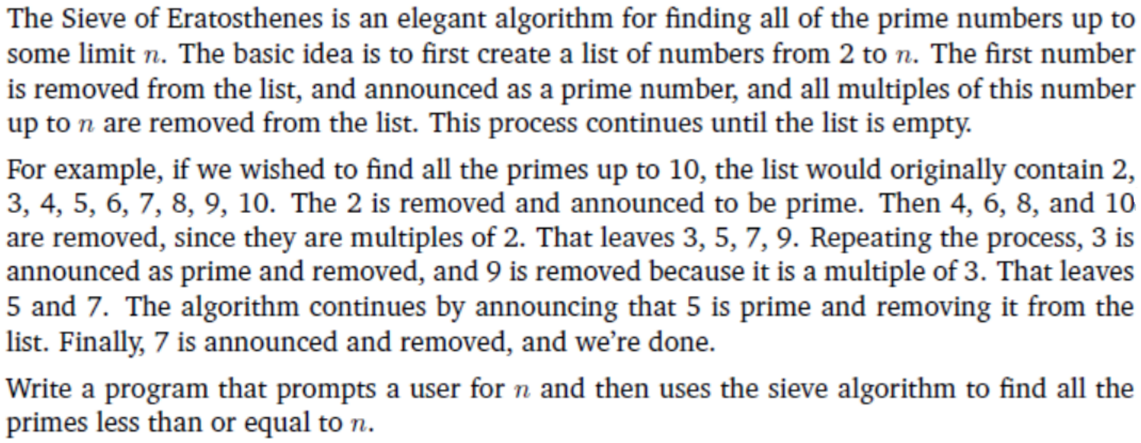
$$y = \frac{57}{23}$$

$$z = \frac{1}{23}$$

08 Extract unique values from a random dictionary.

09 Determine the kth non repeating character in a string

10



The Sieve of Eratosthenes is an elegant algorithm for finding all of the prime numbers up to some limit n. The basic idea is to first create a list of numbers from 2 to n. The first number is removed from the list, and announced as a prime number, and all multiples of this number up to n are removed from the list. This process continues until the list is empty.

For example, if we wished to find all the primes up to 10, the list would originally contain 2, 3, 4, 5, 6, 7, 8, 9, 10. The 2 is removed and announced to be prime. Then 4, 6, 8, and 10 are removed, since they are multiples of 2. That leaves 3, 5, 7, 9. Repeating the process, 3 is announced as prime and removed, and 9 is removed because it is a multiple of 3. That leaves 5 and 7. The algorithm continues by announcing that 5 is prime and removing it from the list. Finally, 7 is announced and removed, and we're done.

Write a program that prompts a user for n and then uses the sieve algorithm to find all the primes less than or equal to n.