Problem Statement - B1

Secret Message

You are trying to pass a sequence of numbers to your friend, but you do not want anyone else to see the contents of the message. What you need is a way to encrypt the message. One of the fastest ways to accomplish this is using a shift cipher. This encryption method takes a key (number) and will shift each digit in your message that many spots. Because the range of each digit is 0-9, the cipher should loop around if shifting the digit by the key causes it to leave this range. For example, the digit 4 with key 3 would be encrypted as 7; the digit 8 with key 5 will be encrypted as 3.

Write a program that takes in an encryption key and the value for 3 digits. The program should then print the encrypted sequence of digits on one line. You can assume that the value of each digit will be 0-9, and you can assume that any key given will be positive. You **cannot assume** that the keys will be less than 10.

Example 1	Example 2
Input: Enter encryption key: 4 **Input:** Enter digit 1: 8 **Input:** Enter digit 2: 2 **Input:** Enter digit 3: 6 **Output:** 260	**Input:** Enter encryption key: 67 **Input:** Enter digit 1: 1 **Input:** Enter digit 2: 7 **Input:** Enter digit 3: 3 **Output:** 840

Solution - B1

Secret Message

```
using System;

namespace ShiftCipherBeginner
{
    class Program
    {
        static void Main(string[] args)
        {
            int[] numbers;

            Console.Write("Enter encryption key: ");
            int key = Convert.ToInt32(Console.ReadLine());
            numbers = new int[3];

            for(int i = 0; i < 3; i++)
            {
                Console.Write($"Enter digit {i}: ");
                numbers[i] = Convert.ToInt32(Console.ReadLine());
            }
            Console.Write("Encrypted: ");
            for (int i = 0; i < 3; i++)
            {
                numbers[i] = (numbers[i] + key);
                while (numbers[i] >= 10) numbers[i] -= 10;
                Console.Write(numbers[i]);
            }
        }
    }
}
```

2 - Beginner - Backyard Overhaul

Problem Statement

This summer, the Griswald family would like to host the bi-weekly neighborhood barbeque. In order to prepare, they want to install brand new two-toned brick edging around their flower bed, but they aren't sure what will look best. They tell the construction company that whatever design they pick will be fine as long as it adheres to the following rules:

- The design must use a different number of red bricks and tan bricks.
- The design cannot use 3 or more consecutive bricks of the same color.

Given a sequence of n colored bricks, output whether or not the sequence conforms to the Griswald family's design rules.

Input

Your program should take the following input:

- An integer n, which represents the number of bricks in the sequence, where $0 < n \leq 10$
- A string of length n containing the colors of each brick (**R** for red and **T** for tan)

Output

Your program should output "True" or "False" indicating that the given grid of bricks conforms or does not conform to the specified rules.

Example

Input	Output
5 RRTTR	True
8 TRRTTRTR	False
4 TRRR	False

b2-backyard.py

```python
# n is not needed for the Python implementation but would be needed if you used arrays in other languages.
n = int(input("Enter the size of the sequence: "))

red = 0
tan = 0
consecutive = 1
last_color = ''
correct = True

row = input('Enter the sequence of bricks: ')

for brick in row:
    if brick == 'R':
        red += 1
    else:
        tan += 1
    if last_color == brick:
        consecutive += 1
    else:
        consecutive = 1
        last_color = brick
    if consecutive >= 3:
        correct = False
if red == tan:
    correct = False
print(correct)

```

3 Beginning — Polynomials
Problem Statement

Every cubic polynomial has exactly 3 roots, though some may not be real, and some may be duplicates. Given 3 roots r_1, r_2, r_3, and a value a, a polynomial having exactly this set of roots may be computed as follows:

$$a(x - r_1)(x - r_2)(x - r_3).$$

For example, if $r_1 = 1$, $r_2 = 2$, $r_3 = -1$, and $a = 1$, the polynomial would be $x^3 - 2x^2 - x + 2$.

Write a program that takes as input a positive 3 real roots (duplicates are allowed) and produces as output the coefficients of the polynomial having these roots, assuming $a = 1$. You must list the coefficients in order from the x^3 term to the constant term.

Example 1:

```
Enter root1: 1
Enter root2: 2
Enter root3: -1

1 -2 -1 2
```

Example 2:

```
Enter root1: 1.5
Enter root2: -1
Enter root3: 1.5

1 -2 -0.75 2.25
```

```
1  // 3 Beginning - Polynomials
2
3  using System;
4  using System.Collections.Generic;
5  using System.Linq;
6  using System.Text;
7  using System.Threading.Tasks;
8
9  namespace PolynomialsBeginning
10 {
11     class Program
12     {
13         static void Main(string[] args)
14         {
15             Console.Write("Enter root1: ");
16             float r1 = Convert.ToSingle(Console.ReadLine());
17             Console.Write("Enter root2: ");
18             float r2 = Convert.ToSingle(Console.ReadLine());
19             Console.Write("Enter root3: ");
20             float r3 = Convert.ToSingle(Console.ReadLine());
21             Console.WriteLine();
22             Console.Write(1 + " ");
23             Console.Write(-r1 - r2 - r3 + " ");
24             Console.Write(r1 * r2 + r1 * r3 + r2 * r3 + " ");
25             Console.WriteLine(-r1 * r2 * r3);
26             Console.ReadLine();
27         }
28     }
29 }
30
```

Beginning Problem Statement B4

Traffic Signals

There are two traffic signals X miles apart on a straight section of highway. Each traffic signal has a red signal for 30 seconds followed by green signal for G seconds. (Ignore the yellow signal. Assume it is part of the end of the green signal). The second traffic signal turns green D seconds after the first signal turns green. The speed limit on the road is 70 miles per hour and the minimum legal speed is 30 miles per hour.

Suppose a car approaches as the first traffic signal turns green. What is minimum legal speed that the car must maintain to go through the second light on the next green signal (the first green signal after the delay)? What is the maximum legal speed?

Indicate if there is not a legal speed that will permit the car to go through the next green signal.

Assume that the car can instantly achieve the appropriate speed and accurately maintain it for the X miles. Distance X is given in miles. Delay D and Green light length G is given in seconds. Output is given in miles per hour. Distances and times are real numbers.

Example 1:	X = 2	Example 2:	X = 3
	D = 60		D = 120
	G = 30		G = 60
Output:	not legal	Output:	Min = 60
			Max = 70

Source Code
B4 Traffic Signals

```
{
    char Q; Q = 'N';
    while (Q == 'N')
    {
        double Max, Min, X, G, D;
        std::cout << "Hello World!\n";
        std::cout << "\nenter distance X ";
        std::cin >> X;
        std::cout << "\nenter delay D ";
        std::cin >> D;
        std::cout << "\nenter green duration G ";
        std::cin >> G;
        Max = (X * 60 * 60) / (D);
        std::cout << "\nMax = " << Max;
        Min = (X * 60 * 60) / (D + G);
        std::cout << "\nMin = " << Min;
        if (Max < 30 || Min > 70) { std::cout << "\nNo legal speed "; }
        else {
            if (Min < 30) { Min = 30; }
            if (Max > 70) { Max = 70; }
            std::cout << "\nMin Speed = " << Min << "\n Max speed = " << Max;
        }
        std::cout << "Quit?";
        std::cin >> Q;
    }
```

5 - Beginner - Dr. Indigo's Zoo

Problem Statement

In their free time, when they're not busy saving the world, Dr. Indigo runs a zoo. Every morning they get up and makes sure that none of the animals have escaped. They have a huge list of all the animals and checks each animal off as they see it but thinks this is really inefficient. They only care about diet of each animal, since animals with the same diet share a cage. There are three cages in the zoo: one for carnivores, one for herbivores, and one for omnivores

Given an integer n, and n lines describing animals, output each the number of animals in each of the cages.

Input

Your program should take the following input:

- An integer n, which represents the total number of animals, where $0 < n < 10$
- n strings, each describing an animal. The last letter in the string will represent that animal's diet (c for carnivore, h for herbivore, and o for omnivore). You may assume that all input will be lower case letters.

Output

Your program should output each type of animal along with their count. Output does not have to be in any particular order.

Example

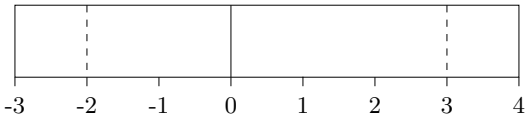
Input	Output
7 african elephant-h white tiger-c gerald the giraffe-h siberian tiger-c tiger-c panda bear-o sloth-o	carnivores: 3 herbivores: 2 omnivores: 2

b5-zoo.py

```
"""
Example input:

7
African elephant-h
White tiger-c
Gerald the Giraffe-h
Siberian tiger-c
Tiger-c
Panda bear-o
Sloth-o

expected output:
elephant: 2
tiger: 3
bear: 1
penguin: 1
"""

# read number of animals
x = int(input("Enter number of animals: "))

# dictionary to store diets, this can be variables as well
animals = {
    'carnivore': 0,
    'herbivore': 0,
    'omnivore': 0,
}

for i in range(x):
    # get last word of line
    diet = input(f'Enter animal {i + 1}: ')[-1]

    # add to dictionary or increment
    if diet == 'o':
        animals['omnivore'] += 1
    elif diet == 'h':
        animals['herbivore'] += 1
    else:
        animals['carnivore'] += 1

# print list
for key in animals:
    print(f"{key}s: {animals[key]}")

```

6 Beginning — Space Race
Problem Statement

A space ship captain is preparing for a race on the course shown to the right. The course lies along a straight line of length 7. The start/finish line is at position 0. The ship must first travel to the right, reaching at least position 3, then going to the left must reach at least position −2 before again reaching the finish line (position 0).



The ship begins at rest. At the beginning of the race, and every second thereafter, the captain can apply an acceleration either to the right (a positive value) or to the left (a negative value). This acceleration lasts for a duration of one second, until the next acceleration is applied. The ship must stay within the course (i.e., at least position −3 and at most position 4) or it will be disqualified.

Write an interactive program that takes as input the accelerations to apply at each second, and after each acceleration is applied for one second, outputs the ship's current position. Each acceleration consists of a floating-point value at least −1 and at most 1. If at any point the ship leaves the bounds of the race course, display a message, "Out of bounds", and stop the simulation. When the ship finishes the course, after displaying the position at the end of the current second, display a message, "Race finished." You may assume that the ship will not cross the finish line, then go out of bounds within the same second (in fact, this is impossible). Use the following formulas to update the position and speed, where t represents the length of the time interval, d represents the net change in position during the time interval, v represents the speed at the beginning of the time interval, v' represents the speed at the end of the time interval, and a represents the acceleration applied:

$$d = vt + at^2/2$$
$$v' = v + at$$

Note that it is possible for the ship to cross one of the goal lines or a course boundary, then re-cross it before the current second elapses. To check for this condition, whenever the old speed and the new speed have opposite signs (i.e., one is positive and the other is negative), use the distance traveled before the ship changes directions ($-v^2/(2a)$).

In the examples that follow, output is shown in italics. Your output should match the output shown to three significant digits.

Example 1:

```
Enter acceleration: 1
0.5
Enter acceleration: 0
1.5
Enter acceleration: 0
2.5
Enter acceleration: -1
3
Enter acceleration: -1
2.5
Enter acceleration: -1
1
Enter acceleration: 1
-0.5
Enter acceleration: 0
-1.5
Enter acceleration: 1
-2
Enter acceleration: 1
-1.5
Enter acceleration: 1
0
Race finished.
```

Example 2:

```
Enter acceleration: 1
0.5
Enter acceleration: -1
1
Enter acceleration: 1
1.5
Enter acceleration: 0.57
2.785
Enter acceleration: -1
3.855
Enter acceleration: -1
Out of bounds.
```

```
// 6 Beginning - Space Race

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace SpaceRaceBeginning
{
    class Program
    {
        static void Main(string[] args)
        {
            float x = 0;
            float v = 0;
            int goals = 0;
            while (goals < 3)
            {
                Console.Write("Enter acceleration: ");
                float a = Convert.ToSingle(Console.ReadLine());
                float d = v + a / 2;
                float px = x + d;
                float newV = v + a;
                if (v * newV < 0)
                {
                    px = x - v * v / (2 * a);
                }
                if (px > 4 || px < -3)
                {
                    Console.WriteLine("Out of bounds.");
                    Console.ReadLine();
                    return;
                }
                if (goals == 0 && px >= 3 || goals == 1 && px <= -2 || goals
                  == 2 && px >= 0)
                {
                    goals++;
                }
                x += d;
                v = newV;
                Console.WriteLine(x);
            }
            Console.WriteLine("Race finished.");
            Console.ReadLine();
        }
    }
}

```