



Table of Contents

1.Dice Game: Sixes Bet.....	2
2.Long Jump.....	2
3.Leap Years.....	2
4.Chess Reward.....	3
5.Guess the Number.....	3
6.Input validation – Registration.....	4
7.Input Validation Loop – IP addresses.....	4
8.Weight Conversions.....	5
9.Validate Internet Domain.....	5
10.Caesar Cipher.....	6

Python Exercises 2b: Repetition Control Structures



1. Dice Game: Sixes Bet

Sixes Bet is an old dice game. The bet is simple: will the player roll a 6 at least once out of four rolls?

Write a Python program to play the game. The program should allow the player to roll the die up to 4 times, display the roll, and check if the player has rolled a six. If the player has rolled a 6, the game is over and the player has won. If the player rolls 4 times and hasn't rolled a 6, s/he loses.

2. Long Jump

In a long jump competition, a jumper takes 6 jumps, and his/her best jump is recorded.

Jump	1	2	3	4	5	6
Distance	8.37	8.43	8.40	8.11	8.31	8.44

Write a program which inputs 6 long jump distances displays the average, maximum and minimum jump, without using a list or other data structure.

3. Leap Years

A year is a *leap year* if it is divisible by 4, unless it is a century year (divisible by 100) that is not divisible by 400. e.g. 1800 and 1900 were not leap years; 1600 and 2000 were. One way to approach this is: a year is a leap year

if it is divisible by 400

OR it is divisible by 4 and not divisible by 100

Write a program which inputs two years, and then determines and displays whether or not each year between them is a leap year.

To check if one number is evenly divisible by another, use the modulo operator, e.g.

```
if year % 4 == 0
```

checks if the value of `year` is evenly divisible by 4.

Test your program with 2000 to 2020, and then test it again with 1890 to 1910.

Python Exercises 2b: Repetition Control Structures



4. Chess Reward

Legend has it that the game of chess was invented by Sissa ben Dahir of the court of King Shiram of India. The king was so impressed with this invention that he promised to give Sissa any reward he wished for.

Being a clever chap, Sissa gave the king a choice: the king could either give him 10,000 rupees, or a payment of wheat based on the 64 square chessboard. To pay Sissa in wheat, the king would need to give him one grain of wheat for the first square, 2 grains of wheat for the second square, 4 for the third, 8 for the fourth, and so on. Each square should have double the amount of wheat as the previous square.

1	2	4	8	16	32	64	128
256	512						
							2^{63}

Now, the kingdom was known for its wheat production, so King Shiram was more than willing to part with some wheat. He thought Sissa a fool to ask for such an insignificant award.

Write a program which calculates and displays:

- the number of grains of wheat on each square,
- and the total number of grains of wheat.

5. Guess the Number

Guess the Number is a simple guessing game. The computer “thinks” of a number, and the player has ten attempts to guess it. If the player guesses the number, s/he wins. Otherwise, the computer tells the player if the guess was too high or too low.

Design and write a program to play *Guess the Number*. The program randomly selects a number between 1 and 100, and then repeatedly (up to 10 times)

- prompts the player to input his/her guess
- checks if it is the right number, it should stop repeating
- otherwise displays a message indicating whether the guess was too high or too low

The program should then display a message indicating whether the player has won or lost.

Python Exercises 2b: Repetition Control Structures



6. Input validation – Registration

Write a program which simulates registering a user account.

The user should be prompted to enter a valid username. A username is valid if it is at most 8 characters and contains no uppercase letters. Use a `while` loop to ensure that a valid username is entered.

The user should then be prompted to enter his/her password, and again to confirm it. Use a `while` loop to ensure that a) the two passwords match, and b) the password is at least 8 characters long.

Once a valid username and password has been entered, the message “Registration Complete” should be displayed.

7. Input Validation Loop – IP addresses

An IP address (version 4) takes the form $w.x.y.z$ where w , x , y and z are numbers in the range 0 – 255, e.g. 192.168.34.10. It would be a mistake to any of the numbers was negative, or greater than 255.

Write a program which prompts the user to input a valid IP address. The program should use a counting loop to input the 4 numbers of the IP address, and use an input validation loop to ensure that the number is valid before proceeding. Once a valid number has been input, it should be added to the IP address.

Hints:

You'll need to input each number as an integer, and check if its between 0 and 255, inclusive.

If this is the first valid number (count is zero), then initialise the ip address string using:

```
ip_address = str(number)
```

where the `str` function converts the number to a string; then increase the count by 1.

For each subsequent valid number, add it to the ip address using:

```
ip_address = ip_address + "." + str(number)
```

or alternatively

```
ip_address += "." + str(number)
```

Python Exercises 2b: Repetition Control Structures



8. Weight Conversions

The conversion from Imperial to Metric weights, and vice versa, are:

1 pound = 0.453 592 37 kilogram

1 kilogram = 2.204 622 621 8 pound

Write a program which repeatedly displays a menu offering to convert from pounds to kilograms, and kilograms to pounds and then processes the users choice until s/he is finished. The program should display a suitable message if the user inputs an invalid choice.

9. Validate Internet Domain

A domain name is an identification string that defines a realm of administrative autonomy, authority or control within the Internet. https://en.wikipedia.org/wiki/Domain_name

Traditionally, a domain name can contain at most 253 characters, and can only consist of letters, numbers, dots . and dashes (hypens) -

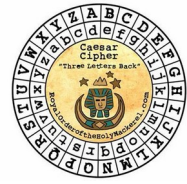
Write a program which inputs and validates a domain name.

Python Exercises 2b: Repetition Control Structures



10. Caesar Cipher

Julius Caesar is credited with a basic substitution cipher in which he shifted each letter of the alphabet along by 3 places. The original message is referred to as the *plaintext*, represented in lowercase characters, and the enciphered message is referred to as the *ciphertext*, represented in uppercase characters.



a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

So a is enciphered as D, b is enciphered as E, ... The letters at the end of the alphabet (x, y and z) wrap-around to be substituted by the first three.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Write a program which inputs a plaintext message, enciphers it using the Caesar Cipher, and then displays the ciphertext.

The logic is as follows:

Convert the plaintext to lowercase

Take each character in the plaintext one at a time

If the character is a letter, convert it into a number representing its position in the alphabet

Add 3 to this number

Find the letter corresponding to this number and add it to the ciphertext

Otherwise add the character on to the ciphertext

Display the ciphertext

Sample Output

Program to encipher a message using the Caesar Cipher

Enter the message: The Celts are attacking!

The enciphered message is: WKH FHOWV DUH DWWDFNLQJ!