SILVER LEVEL TASKS

# Task - Modulus

Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included).

The numbers obtained should be printed in a comma-separated sequence on a single line.

## Functions to use

[print](https://www.w3schools.com/python/ref_func_print.asp)

[append](https://www.w3schools.com/python/ref_list_append.asp)

[str](https://www.w3schools.com/python/ref_func_str.asp)

[for](https://www.w3schools.com/python/python_for_loops.asp)

[range](https://www.w3schools.com/python/ref_func_range.asp)

[%](https://www.w3schools.com/python/python_operators.asp)

[if](https://www.w3schools.com/python/python_conditions.asp)

and

[join](https://www.w3schools.com/python/ref_string_join.asp)

## Save

Save your file as **bysevennotfive.py**

# Task - Factorials

## Write a program which can compute the factorial of a given number.

## The results should be printed in a comma-separated sequence on a single line.

## Suppose the following input is supplied to the program:

## 8

## Then, the output should be:

## 40320

## Functions/Operators to use

[print](https://www.w3schools.com/python/ref_func_print.asp)

[def](https://www.w3schools.com/python/python_functions.asp)

[input](https://www.w3schools.com/python/ref_func_input.asp)

[if](https://www.w3schools.com/python/python_conditions.asp)

[return](https://www.w3schools.com/python/ref_keyword_return.asp)

## Save

Save your file as **factorials.py**

# Task – Dict in Python

With a given integral number n, write a program to generate a dictionary that contains (i, i\*i) such that is an integral number between 1 and n (both included). and then the program should print the dictionary.

Suppose the following input is supplied to the program:

8

Then, the output should be:

{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

## Functions/Operators to use

[print](https://www.w3schools.com/python/ref_func_print.asp)

dict

[for](https://www.w3schools.com/python/python_for_loops.asp)

[range](https://www.w3schools.com/python/ref_func_range.asp)

[int](https://www.w3schools.com/python/ref_func_int.asp)

[input](https://www.w3schools.com/python/ref_func_input.asp)

## Save

Save your file as **usingdict.py**

# Task – Tuples

Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number.

Suppose the following input is supplied to the program:

34,67,55,33,12,98

Then, the output should be:

['34', '67', '55', '33', '12', '98']

('34', '67', '55', '33', '12', '98')

## Functions/Operators to use

[input](https://www.w3schools.com/python/ref_func_input.asp)

[upper](https://www.w3schools.com/python/ref_string_upper.asp)

[split](https://www.w3schools.com/python/ref_string_split.asp)

[tuple](https://www.w3schools.com/python/python_tuples.asp)

[print](https://www.w3schools.com/python/ref_func_print.asp)

## Save

Save your file as **startingwithtuples.py**

# Task – Alphabetical sort

Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically.

Suppose the following input is supplied to the program:

without,hello,bag,world

Then, the output should be:

bag,hello,without,world

## Functions/Operators to use

[class](https://www.w3schools.com/python/python_classes.asp)

[def](https://www.w3schools.com/python/python_functions.asp)

[input](https://www.w3schools.com/python/ref_func_input.asp)

[upper](https://www.w3schools.com/python/ref_string_upper.asp)

[print](https://www.w3schools.com/python/ref_func_print.asp)

## Save

Save your file as **alphasort.py**

# Task – Card shuffle

Create an array or arrays that contain all the cards within a deck of cards

Display them sorted

Display them 'shuffled' using the random function.

## Save

Save your file as **cardshuffle.py**

# Task – Snap card game

Copy your **code** from the card shuffle task.

Create the game based on the **traditional rules** below:

## Rules

The player on the dealer’s left turns over the top card of his pile and puts it face up starting a pile of cards next to their face down cards. The next player to the left does the same and so on around the table.

When someone turns up a card that matches a card already face up on another player’s pile, the first person to notice the two matched cards calls out “Snap!” and wins both piles. This player adds the cards to the bottom of their face-down pile.

When two players shout “Snap!” at the same time, the two piles are combined and placed in the center of the table face up. These cards form a “Snap Pot.” Play continues where it left off with the player to the left of the last player who turned over a card. If a player spots a card that matches the card on top of the Snap Pot, they shout “Snap Pot!” and win all of those cards. During the game, if a player runs out of cards in their face-down pile, the cards in the face up pile are turned down and the player continues to play.

# Task – 21 Card Game

Copy your code from the **card shuffle** task.

Create the game based on the **traditional card** rules below:

## Rules

* Aim of the 21 Card Game is to get 21 or as close to as possible.
* Number cards have their face value, jacks, kings and queens are worth 10. Ace can be either 1 or 11 and the player who holds the ace gets to choose the value of the card.
* The dealer and all other players have two cards. With the exception of the dealer the players have their cards face up. The dealer has one card up and one card face down.
* The dealer goes to each player one at a time. The player needs to decide if they want another card (hit) or will sit on what they have. You can have as many cards as you like as long as you don’t go over 21.
* The dealer does this with every player. Players are not competing against each other, but against the dealer.
* The dealer then turns over their other card and needs to decide what to do. If the dealer has 16 or under then they must take another card.
* If the dealer has 21 (Ace and a ten value card) the dealer wins.
* If the dealer goes bust then everyone else wins.
* We reshuffle the deck of cards after every game.