

Task1:

Create a set of integers and print out the size of the set.

Example:

my\_set = {1, 2, 3, 4, 5}

output:

Size of the set: 5

\_ \_ \_ \_ \_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_ \_ \_ \_\_\_ \_\_ \_ \_ \_  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

Task2:

Write a program that takes in two sets of integers and prints out the

intersection and union of the sets.

Example:

set1 = {1, 2, 3, 4, 5}

set2 = {4, 5, 6, 7, 8}

output:

Intersection of the sets: {4, 5}

Union of the sets: {1, 2, 3, 4, 5, 6, 7, 8}

\_ \_ \_ \_ \_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_ \_ \_ \_\_\_ \_\_ \_ \_ \_  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

Task3:

Create a set of strings and print out all the strings in the set.

Examples:

mySet = {"apple", "banana", "orange", "pear"}

output:

apple

banana

orange

pear

\_ \_ \_ \_ \_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_ \_ \_ \_\_\_ \_\_ \_ \_ \_  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

Task4:

Create two sets of strings and print out the difference between them (i.e., the items that are in the first set but not the second).

Example:

set1 = {"apple", "banana", "orange", "pear"}

set2 = {"banana", "pear"}

output:

Difference between the sets: {'apple', 'orange'}

\_ \_ \_ \_ \_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_ \_ \_ \_\_\_ \_\_ \_ \_ \_  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

Task5:

Write a program that takes in a list of integers and removes any duplicates in

the list.

Example:

my\_list = [1, 2, 3, 3, 4, 4, 5, 6, 6, 6]

output:

Original list: [1, 2, 3, 3, 4, 4, 5, 6, 6, 6]

List with duplicates removed: [1, 2, 3, 4, 5, 6]

\_ \_ \_ \_ \_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_ \_ \_ \_\_\_ \_\_ \_ \_ \_  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

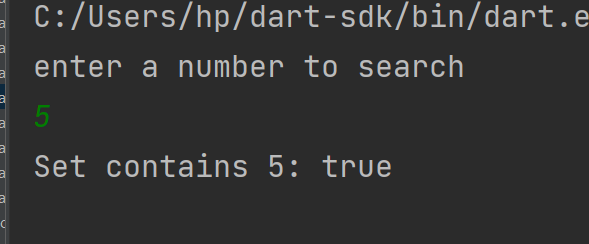
Task6:

Create a set of integers and use check if a specific integer is in the set.

Example:

intSet = {1, 2, 3, 4, 5};

output:

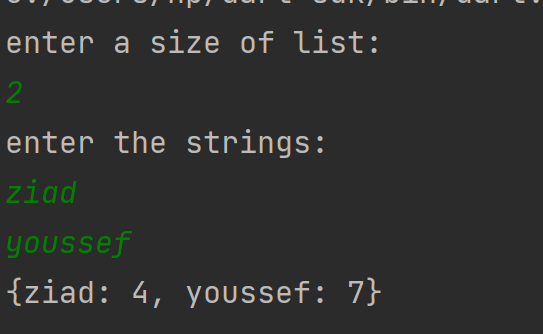




Task 8:

Write a program that takes in a list of strings and returns a map where the keys are the strings and the values are the lengths of the strings.

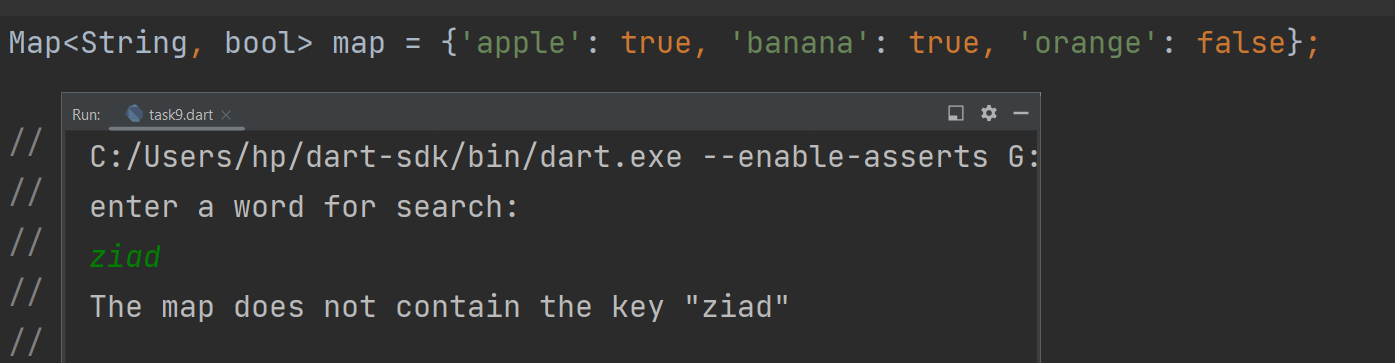
Example:

\_ \_ \_ \_ \_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_ \_ \_ \_\_\_ \_\_ \_ \_ \_  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

Task 9 :

Create a map that maps strings to booleans and check if a specific key is in the map.

Example 1 :



Example 2 :

// take the strings and Booleans from users

Text

Description automatically generated \_ \_ \_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_ \_ \_ \_\_\_ \_\_ \_ \_ \_  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

Task 10 :

Create a map that maps strings to lists of integers and print out the contents of the map using foreach.

Example:

Output:

//hint: map contains strings 'even' 'odd' 'prime' as keys and list<int> as values

even: [2, 4, 6, 8]

odd: [1, 3, 5, 7, 9]

prime: [2, 3, 5, 7]\_



\_ \_ \_ \_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_ \_ \_ \_\_\_ \_\_ \_ \_ \_  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

Task11: BONUS

Write a program that takes in a list of strings and create a map where the keys are the first letters of the strings and the values are lists of the strings that start with that letter.

List<String> strings = [  
 'apple', 'banana', 'avocado', 'kiwi', 'pear'  
];

Output:

\_ \_ \_ \_ \_ \_ \_ \_\_ \_ \_ \_\_ \_ \_ \_ \_ \_ \_\_\_ \_\_ \_ \_ \_  \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

A picture containing text, clipart

Description automatically generated

By : Ziad Elsaadany