1) Given a two list. Create a third list by picking an odd-index element fr list and even index elements from second.

Input:

```
listOne = [3, 6, 9, 12, 15, 18, 21]

listTwo = [4, 8, 12, 16, 20, 24, 28]
```

Output:

```
[6, 12, 18, 4, 12, 20, 28]
```

2) Given a list slice it into a 3 equal chunks and reverse each list

Input:

```
sampleList = [11, 45, 8, 23, 14, 12, 78, 45, 89]
```

Output:

```
[8, 45, 11]

[12, 14, 23]

[89, 45, 78]
```

3) Given a list iterate it and count the occurrence of each element and c dictionary to show the count of each element

```
Original list  [11, 45, 8, 11, 23, 45, 23, 45, 89]

Printing count of each item  {11: 2, 45: 3, 8: 1, 23: 2, 89: 1}
```

[Type here]

4) Given a two list of equal size create a set such that it shows the elem both lists in the pair

```
First List  [2, 3, 4, 5, 6, 7, 8]

Second List  [4, 9, 16, 25, 36, 49, 64]

Result is  {(6, 36), (8, 64), (4, 16), (5, 25), (3, 9), (7, 49), (2, 4)}
```

5) Iterate a given list and Check if a given element already exists in a di a key's value if not delete it from the list

Input :

```
rollNumber = [47, 64, 69, 37, 76, 83, 95, 97]

sampleDict ={'Jhon':47, 'Emma':69, 'Kelly':76, 'Jason':97}
```

Output:

```
[47, 69, 76, 97]
```

6) Given a dictionary get all values from the dictionary and add it in a lis add duplicates

Input:

```
speed ={'jan':47, 'feb':52, 'march':47, 'April':44, 'May':52, 'June':53, 'july':54, '
'Sept':54}
```

Output:

[Type here]

[47, 52, 44, 53, 54]

7) Remove duplicate from a list and create a tuple and find the minimun maximum number

Input:

```
sampleList = [87, 45, 41, 65, 94, 41, 99, 94]
```

Output:

```
unique items [87, 45, 41, 65, 99]

tuple (87, 45, 41, 65, 99)

min: 41

max: 99
```

8) You are given a list of integers. Your task is to do the following:

1) Replace all integers that are evenly divisible by 3 with "fizz"

2) Replace all integers divisible by 5 with "buzz"

3) Replace all integers divisible by both 3 and 5 with "fizzbuzz"

Input: numbers = [45, 22, 14, 65, 97, 72]

9) Sort the following list of complex data type in ascending order of age of a

Input: animals = [{'type': 'penguin', 'name': 'Stephanie', 'age': 8},

{'type': 'elephant', 'name': 'Devon', 'age': 3},

{'type': 'puma', 'name': 'Moe', 'age': 5}]

[Type here]

10) You have a function named get_random_word(). It will always return selection from a small set of words:

```
import random

all_words = "all the words in the world".split()

def get_random_word():

    return random.choice(all_words)
```

Write a function get_unique_words() which repeatedly calls get_random_wo 1000 random words and then return a data structure containing every uniqu

11) Given a dictionary cowboy .

```
cowboy = {'age': 32, 'horse': 'mustang', 'hat_size': 'large'}
```

Write a code to get the 'name' of cowboy. If key is absent, set cowboy['name Man with No Name' and return the new value.

12) Write a program that reads integers from the user and stores them in program should continue reading values until the user enters 0. Then it shou all of the values entered by the user (except for the 0) in order from smalles with one value appearing on each line. Use either the sort method or the so to sort the list.

13) Create a program that reads integers from the user until a blank line i Once all of the integers have been read your program should display all of t numbers, followed by all of the zeros, followed by all of the positive numbers each group the numbers should be displayed in the same order that they we by the user. For example, if the user enters the values 3, -4, 1, 0, -1, 0, and program should output the values -4, -1, -2, 0, 0, 3, and 1. Your program sh each value on its own line.

[Type here]