**Name :** Makwana Vishal Raghavbhai

**Class** : FYMCA (SAM-2)

**Div :** B

**Roll No:** 113

**Paper Number :** 204

**Subject** : Python Programming Language

**Assignment Number :** 3

**Dept. of Computer Science,**

**Veer Narmad South Gujarat University, Surat.**

**M.C.A. 2nd Semester (2020-2021)**

**Paper 204: Python Programming Language**

## Practical Sheet - 3

**Q.1. Write a Python program to create a list of numbers by taking input from the user and then remove the duplicates from the list. You can take input of non-zero numbers, with an appropriate prompt, from the user until the user enters a zero to create the list assuming that the numbers are non-zero.**

**Sample Input: [10, 34, 18, 10, 12, 34, 18, 20, 25, 20]**

**Output: [10, 34, 18, 12, 20, 25]**

```
numbers = []
unique = []
while True:
    n = int(input("enter the number for exit(nagative
                  Number)"))
    if n <= 0:
        break
    else:
        numbers.append(n)


for i in numbers:
    if not i in unique:
        unique.append(i)
    else:
        pass

print(f"Before removing element into the list are :
      {numbers}")
print(f"After removing element into the list arr : {unique}")
```

**Q.2. Write a Python program to create a list of lists of numbers (i.e. each element of the list is a list of numbers e.g. [[1, 2], [3, 2, 5], [1], [5, 3, 6, 7]]. Then generate a list from the given list where each element of the list is the length of each list in the given list. i.e. for the given example the output should be [2, 3, 1, 4]. You can take input of non-zero numbers, with an appropriate prompt, from the user until the user enters a zero to create the list assuming that the numbers are non-zero.**

**Sample Input: [[1, 2], [3, 2, 5], [1], [5, 3, 6, 7]]**

**Output: [2, 3, 1, 4]**

```
length = []
numbers = []

def fun():
    templist = []
    while True:
        n = int(input("enter a number into list"))
        if n < 0:
            break
        else:
            templist.append(n)
            print(templist)
    return templist

while True:
    exit = input("for input press a (enter) and for exit type
                  (exit)")
    if exit.lower() == 'exit':
        break
    else:
        numbers.append(fun())


for i in numbers:
    length.append(len(i))

print(f"The list is : {numbers}")
print(f"length of each list in the given list : {length}")
```

**Q.3. Write a Python program to create a list of numbers by taking input from the user. Split this list into two tuples where one tuple contains odd numbers and the other tuple contains even numbers from the list. You can take input of non-zero numbers, with an appropriate prompt, from the user until the user enters a zero to create the list assuming that the numbers are non-zero.**

**Sample Input: [10, 12, 13, 90, 43, 32, 30, 11]**

**Output:**

**List of Odd Numbers: [13, 43, 11]**

**List of Even Numbers: [10, 12, 90, 32, 30]**

```
numbers = []
odd_tup = ()
even_tup = ()
while True:
    n = int(input("enter the number  for exit enter(0)"))
    if n == 0:
        break
    else:
        numbers.append(n)

for i in numbers:
    if i % 2 == 0:
        e = list(even_tup)
        e.append(i)
        even_tup = tuple(e)

    else:
        o = list(odd_tup)
        o.append(i)
        odd_tup = tuple(o)

print(f"list is : {numbers}")
print(f"Odd mumbers tuple : {odd_tup}")
print(f"Even mumbers tuple : {even_tup}")
```

**Q.4. Write a Python program to create a list of elements of any data type (mixed data type, i.e. some elements maybe of type int, some elements of type float and some elements of type string). Splitthis list into three tuples containing elements of same data type (i.e. 1st tuple of integers only,2nd tuple of float only and 3rd tuple of strings only). Take input from the user to create the list.**

**Sample Input: [25, 4.5, 'Hello', 90, 20, 7.5, 9.25, 'World']**

**Output:**

**List of Integers: [25, 90, 20]**

**List of Float Values: [4.5, 7.5, 9.25]**

**List of Strings: ['Hello', 'World']**

```
lst = [1,2,'vishal','jay',3.6,8.3,'v']
int_tup = ()
float_tup = ()
string_tup = ()
char_tup = ()
print(lst)

for i in lst:
    if type(i) == int:
        it = list(int_tup)
        it.append(i)
        int_tup = tuple(it)
    elif type(i) == float:
        ft = list(float_tup)
        ft.append(i)
        float_tup = tuple(ft)
    elif type(i) == str:
        if len(i) == 1:
            ct = list(char_tup)
            ct.append(i)
            char_tup = tuple(ct)
        else:
            st = list(string_tup)
            st.append(i)
            string_tup = tuple(st)


print(f"Integer are : {int_tup}")
print(f"Float are : {float_tup}")
print(f"Character are : {char_tup}")
print(f"String are : {string_tup}")
```

**Q.5. Write a Python program to create a dictionary of student data by taking input from the user,where each student data contains Rollno (to be considered as key), and marks in 3 subjects (to be considered as list of values). e.g. {1 : [45, 40, 35], 2 : [41, 38, 39], 3 : [35, 30, 37]}. Prepare mark list in the following format:**

| Roll No. | Mark-1 | Mark-2 | Mark-3 | Total |
|---|---|---|---|---|
| 1 | 45 | 40 | 35 | 120 |

```
n = int(input("How many Records you want to add"))

d = {}

for i in range(n):
    key = int(input("ener a roll number"))
    mark1 = int(input("enter a mark1"))
    mark2 = int(input("enter a mark2"))
    mark3 = int(input("enter a mark3"))

    d[key] = [mark1,mark2,mark3]
print(d)
print("Roll No".center(20," "),end="")
print("Mark 1".center(20," "),end="")
print("Mark 2".center(20," "),end="")
print("Mark 3".center(20," "),end="")
print("Sum ".center(20," "))

for key,val in d.items():
    print(str(key).center(20," "),end="")
    for i in val:
        print(str(i).center(20," "),end="")
    print(str(sum(val)).center(20," "))
```

**Q.6. Write a Python program to take input of a string from the user and then create a dictionary**

**containing each character of the string along with their frequencies. (e.g. if the string is 'banana'**

**then output should be {'b': 1, 'a': 3, 'n': 2}.**

```
count_freq = {}
s = input("input a string ")
for i in s:
    if i in count_freq:
        count_freq[i] += 1
    else:
        count_freq[i] = 1

print(str(count_freq))
```

**#Another Method**

```
from collections import Counter

s = input("input a string ")

s_count = Counter(s)

print(s_count)
```

**Q.7. Write a Python program to create a list of strings by taking input from the user and then create a dictionary containing each string along with their frequencies. (e.g. if the list is [‘apple’,‘banana’, ‘fig’, ‘apple’, ‘fig’, ‘banana’, ‘grapes’, ‘fig’, ‘grapes’, ‘apple’] then output should be {'apple': 3, 'banana': 2, 'fig': 3, 'grapes': 2}.**

```
count_freq = {}
words = []
while True:
    s = str(input("input a string for exit (0)"))
    if s.lower() == 'exit':
        break
    else:
        words.append(s)


print("\n\n\n")
print(words)
w_freq = [words.count(w) for w in words]
print(dict(zip(words,w_freq)))
```

```
# Another Method

for w in words:
    count_freq[w] = count_freq.setdefault(w,0) + 1
print(count_freq)
```

**Q.8. Write a Python program to input a string that is a list of words separated by commas. Construct**

**a dictionary that contains all these words as keys and their frequencies as values. Then display**

**the words with their quantities.**

```
s = input("enter a key that are saperated by commas")

words = s.split(',')
count_freq = {}

print("\n\n\n")
print(words)
w_freq = [words.count(w) for w in words]
print(dict(zip(words,w_freq)))
```

```
# Another Method

for w in words:
    count_freq[w] = count_freq.setdefault(w,0) + 1
print(count_freq)
```

**Q.9. Consider a very small dictionary that contains the translations of English words to Dutch as**

**shown below:**

**english_dutch = {"last" : "laatst", "week" : "week", "the" : "de", "royal" : "koninklijk","festival" : "feest", "hall" : "hal",}**

**Write a program that uses this dictionary to create a word-for-word translation of a sentence to be taken as an input from the user. A word for which you cannot find a translation, you canleave "as is." The dictionary is supposed to be used case-insensitively, but your translation may consist of all lower-case words. It is nice if you leave punctuation in the translation, but if you take it out, that is acceptable (as leaving punctuation in is quite a bit of work, and does not really have anything to do with dictionaries – besides, leaving punctuation in is much easier to do once you have learned about regular expressions).**

```
english_dutch = {"last" : "laatst", "week" : "week", "the" :
"de", "royal" : "koninklijk","festival" : "feest", "hall" :
"hal}

sentence = "last week the  royal festival hall"

dutch = ''
words = list(sentence.split())

for w in words:
    w = w.lower()
    if w in english_dutch.keys():
        dutch = dutch + english_dutch[w]+' '
    else:
        dutch = dutch + w +' '

print(f"English sentence is : {sentence}")
print(f"dutch sentence is : {dutch}")
```

**Q.10. Consider the following list of movies. For each movie it also shows list of ratings. Write a Python program to convert it in such a way that it stores all this data in one dictionary, then use the dictionary to print the average rating for each movie, rounded to one decimal.**

**movies = ["Where Eagle's Dare", "Enter the Dragon", "Iron Fist", "Fist of Fury"]**

**dare_ratings = [ 9, 10, 9.5, 8.5, 3, 7.5, 8] for the movie "Where Eagle's Dare"**

**dragon_ratings = [ 10, 10, 0, 9, 1, 8, 7.5, 8, 6, 9 ] for the movie "Enter the Dragon"**

**fist_ratings = [ 7, 6, 5 ] for the movie "Iron Fist"**

**fury_ratings = [ 6, 5, 6, 6 ] for the movie "Fist of Fury"**

**Output:**

**Average ratings of the movie "Where Eagle's Dare" is 7.9**

**Average ratings of the movie "Enter the Dragon" is 6.9**

**Average ratings of the movie "Iron Fist" is 6**

**Average ratings of the movie "Fist of Fury" is 5.8**

```
movies = ["Where Eagle's Dare", "Enter the Dragon", "Iron
Fist", "Fist of Fury"]
dare_ratings = [ 9, 10, 9.5, 8.5, 3, 7.5, 8]
dragon_ratings = [ 10, 10, 0, 9, 1, 8, 7.5, 8, 6, 9 ]
fist_ratings = [ 7, 6, 5 ]
fury_ratings = [ 6, 5, 6, 6 ]


v_list =
[dare_ratings,dragon_ratings,fist_ratings,fury_ratings]

d = dict(zip(movies,v_list))

for key,val in d.items():
    print(f'Average rating of movie "{key}" is ',end="")
    print(round(sum(val)/len(val),1))
```

**********