Q-1

1 What will be the output of the following Python code?

```
names1 = ['Amir', 'Bear', 'Charlton', 'Daman']
names2 = names1
names3 = names1
names2[0] = 'Alice'
names3[1] = 'Bob'
sum = 0
for ls in (names1, names2, names3):
   if ls[0] == 'Alice':
      sum += 1
   if ls[1] == 'Bob':
      sum += 10
print(sum)
```

(A). 12 (B). 32 (C). 20
(D). 33 (E). 22 (F). 25
(G). Error

2 What will be the output of the following Python code?

```
def tup(T):
   print(T[T.index(5)], end = " ")
   print(T[T[T[6]-3]-6])
T = (1, 2, 3, 4, 5, 6, 7, 8)
tup(T)
```

(A). 5 IndexError (B). 4 0 (C). SyntaxError
(D). 8 5 (E). 5 8 (F). 4 5
(G). 5 7

3 What will be the output of the following Python code?

```
l= [1, "m", ["a", {1: [1,2,3]}]]
t= {(1,2,3): (5)}
s= (l[2][1][1], t[(1,2,3)])
print(s)
```

(A). a5 (B). ([1, 2, 3], 5) (C). 10
(D). 7 (E). ([1, 2, 3]) (F). (1, 5)
(G). Error

4 What will be the output of the following Python code?

```
def f1(*m):
  sum1=len(m)
  for i in m:
     sum1+=i
  return sum1
x=f1(1,2,3,(4,)==(4,),(5,)==(5))
print(x)
```

(A). 11 (B). 13 (C). 15
(D). 10 (E). 14 (F). 12
(G). Error

5 What is the output of the following code?

```
a="Hello Welcome to the Python"
```

```
print(a.find("z"))
print(a.index("z"))
```

(A). -1
ValueError
(B). -1
SyntaxError
(C). ValueError
-1
(D). SyntaxError
(E). -1
-1
(F). ValueError
(G). 1
-1

6 What will be the output of the following Python code?

```
D = {1: 'one', 'two': 2, 'three': 3}
D[1] = D[2]
D['two'] = D['three']
print(D[D[D['three']]])
```

(A). two (B). three (C). one
(D). 1 (E). 2 (F). TypeError
(G). KeyError

7 What will be the output of the following Python code?

```
import functools
seq = [1, 2, 3, 4, 5]
result = functools.reduce(lambda x, y: x * y, list(filter(lambda x: x % 2 == 0, seq)))
print(result)
```

(A). 120 (B). 15 (C). 8
(D). 60 (E). 30 (F). 4
(G). Error

8 What will be the output of the following Python code?

```
t=(1,2,4,3,6,8,4)
t[1:-1:-1]
```

(A). (2,4,3,6,8) (B). (2,1) (C). ( )
(D). (4,8) (E). (8,6,3,4,2) (F). (4,1)
(G). IndexError

9 What will be the output of the following Python code?

```
import functools
squares = lambda x: list(map(lambda a: a + a, x))
result = functools.reduce(lambda x, y: x + y, squares([1, 2, 3, 4, 5]))
print(result)
```

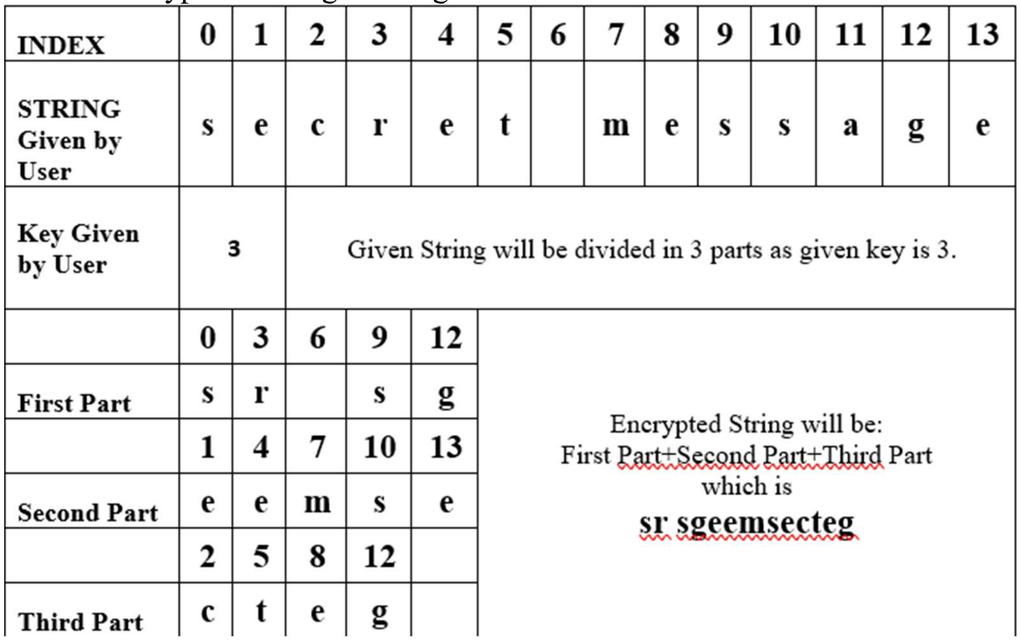
(A). 15 (B). 30 (C). 20
(D). 25 (E). 21 (F). 50
(G). 55

10 What will be the output of the following Python code?

```
l=[1,2,(5)]
l[2]=7
print(l)
```

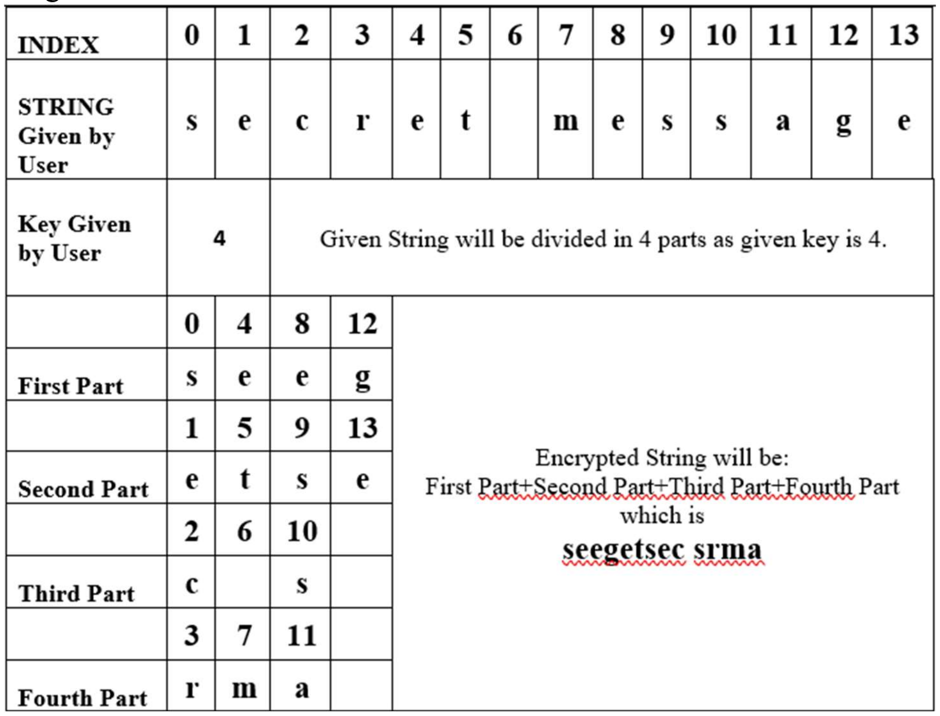
(A). [1, 7, (5)] (B). [1, 7] (C). [1 ,7, 5]
(D). Error (E). 7 (F). [7]
(G). [1, 2, 7]

Q-2 One of the ways to encrypt a string is by rearranging its characters by certain rules, they are broken up by threes, fours or something larger. For instance, in the case of threes, the string ‘secret message’ would be broken into three groups. The first group is sr sg, the characters at indices 0, 3, 6, 9 and 12. The second group is eemse, the characters at indices 1, 4, 7, 10, and 13. The last group is ctea, the characters at indices 2, 5, 8, and 11. The encrypted message is sr sgeemsectea.



INDEX	0	1	2	3	4	5	6	7	8	9	10	11	12	13
STRING Given by User	s	e	c	r	e	t		m	e	s	s	a	g	e
Key Given by User	3		Given String will be divided in 3 parts as given key is 3.											
	0	3	6	9	12	Encrypted String will be: First Part+Second Part+Third Part which is **sr sgeemsecteg**								
First Part	s	r		s	g									
	1	4	7	10	13									
Second Part	e	e	m	s	e									
	2	5	8	12										
Third Part	c	t	e	g										

If the string ‘secret message’ would be broken into four groups. The first group is seeg, the characters at indices 0, 4, 8 and 12. The second group is etse, the characters at indices 1, 5, 9 and 13. The third group is c s, the characters at indices 2, 6 and 10. The fourth group is rma, the characters at indices 3, 7 and 11. The encrypted message is seegetsec srma.



INDEX	0	1	2	3	4	5	6	7	8	9	10	11	12	13
STRING Given by User	s	e	c	r	e	t		m	e	s	s	a	g	e
Key Given by User	4		Given String will be divided in 4 parts as given key is 4.											
	0	4	8	12	Encrypted String will be: First Part+Second Part+Third Part+Fourth Part which is **seegetsec srma**									
First Part	s	e	e	g										
	1	5	9	13										
Second Part	e	t	s	e										
	2	6	10											
Third Part	c		s											
	3	7	11											
Fourth Part	r	m	a											

(A). Write a program that asks the user for a string, and an integer determining whether to break things up by threes, fours, or whatever user inputs. Encrypt the string using above method.

For example,
Input message: This is python, a programming language

Input Key: 4
Output Encrypted Message: T poaomgnghiyn gm geist,prilus h ranaa

Input message: This is python, a programming language
Input Key: 7
Output Message: T ,ggahp r giyaalest ma hpmniorigsnonu

(B). If you get a message which is encoded by the method above then, Write a decryption program for the general case. Taking input of any encrypted string from user with key number used while breaking message apart during encryption.

For example,
Input Encrypted message: Hloe gl o sogrilw g epntstfii o yotay hee nnh aoiortiimreegehrun nhnse ne
Input Key used during encryption: 5
Output Decrypted Message: Hi hello how are you going to learn python in this semester of engineering

Input Encrypted message: Ig ntot oopid ys lt dehaaao yrn
Input Key used during encryption: 8
Output Decrypted Message: It is a good day to learn python

Input Encrypted message: istemoaa!t   e ym ntt p eiohitlgs
Input Key used during encryption: 4
Output Decrypted Message: it is not the time to play games!

Q-3 **Part -1**
There are n children standing in a line. Each child is assigned a rating value given in the integer list of ratings.

You are giving candies to these children subjected to the following requirements:

- Each child must have at least one candy.
- Children with a higher rating get more candies than their neighbors.
- Return the minimum number of candies you need to have to distribute the candies to the children.

For Example:
(1).
Input:
n = 10
ranks = [2, 4, 1, 6, 1, 7, 8, 9, 2, 1]
Output:
Min. number of candies required to distribute: 19

Explanation:
Child 1 (rating 2): 1 candy
Child 2 (rating 4): 2 candies
Child 3 (rating 1): 1 candy
Child 4 (rating 6): 3 candies
Child 5 (rating 1): 1 candy

Child 6 (rating 7): 2 candies
Child 7 (rating 8): 3 candies
Child 8 (rating 9): 4 candies
Child 9 (rating 2): 2 candies
Child 10 (rating 1): 1 candy

(2).
Input:
n = 5
ranks = [9,8,7,6,7]
Output:
Min. number of candies required to distribute: 12

Explanation:
Child 1 (rank 9): 4 candies
Child 2 (rank 8): 3 candies
Child 3 (rank 7): 2 candies
Child 4 (rank 6): 1 candy
Child 5 (rank 7): 2 candies

(3).
Input:
n = 5
ranks = [2,4,5,7,5]
Output:
Min. number of candies required to distribute: 11

Explanation:
Child 1 (rank 1): 1 candy
Child 2 (rank 2): 2 candies
Child 3 (rank 3): 3 candies
Child 4 (rank 4): 4 candies
Child 5 (rank 1): 1 candy

Part-2
Write a program that converts Roman numerals into ordinary numbers. Roman numerals are represented by the following symbols and their respective values:

- M = 1000
- D = 500
- C = 100
- L = 50
- X = 10
- V = 5
- I = 1

Additionally, special cases like IV (4), XL (40) etc. should also be considered. The program should take a Roman numeral as input and output its equivalent ordinary number.

Roman Numerals: 1 - 1000

I	V	X	L	C	D	M
1	5	10	50	100	500	1000

1	I
2	II
3	III
4	IV
5	V
6	VI
7	VII
8	VIII
9	IX
10	X

11	X1
20	XX
30	XXX
40	XL
50	L
60	LX
70	LXX
80	LXXX
90	XC
100	C

200	CC
300	CCC
400	CD
500	D
600	DC
700	DCC
800	DCCC
900	CM
1000	M
1001	MI

For Example:
Input:
Enter Roman Number: CMXCIX
Output:
Output Ordinary Number: 999

Input:
Enter Roman Number: DCLXXVI
Output:
Output Ordinary Number: 676

Input:
Enter Roman Number: MMMCMLXXXIX
Output:
Output Ordinary Number: 3989

Library Book Management and Analysis System

Question:
Write a Python program to manage and analyze a small library's book database using strings, lists, and dictionaries.

Static Input:

Use the following list of strings as input data (**no user input required**):

```
books_data = [
    "Inferno,Dan Brown,Thriller,450,4.5",
    "Atomic Habits,James Clear,SelfHelp,550,4.8",
    "Ikigai,Hector Garcia,SelfHelp,300,4.2",
    "Dune,Frank Herbert,ScienceFiction,600,4.6",
    "1984,George Orwell,ScienceFiction,500,4.7",
    "Sapiens,Yuval Noah Harari,History,650,4.9",
    "The Hobbit,J.R.R. Tolkien,Fantasy,480,4.8",
    "Becoming,Michelle Obama,Biography,700,4.7",
    "The Alchemist,Paulo Coelho,Fiction,350,4.3",
    "To Kill a Mockingbird,Harper Lee,Fiction,400,4.6"
]
```

Your Task:

1. **Convert** the above list of strings into a **dictionary**, where:
    - **Key = Book title**
    - **Value = Another dictionary**
    - {"Author": ..., "Genre": ..., "Price": ..., "Rating": ...}
2. **Perform the following analyses:**
   a. Display the **average price** of all books.
   b. Display the **highest-rated book** (title, author, rating).
   c. **Group books by genre** and show the **count of books** in each genre.
   d. Display all **ScienceFiction** books **sorted by rating (descending)**.
   e. Display the **most expensive book in each genre**.

Expected Output:

```
Average Price of all books: 498.0
```

```
Highest Rated Book: Sapiens by Yuval Noah Harari (Rating: 4.9)

Books by Genre:
Thriller: 1
SelfHelp: 2
ScienceFiction: 2
History: 1
Fantasy: 1
Biography: 1
Fiction: 2

ScienceFiction Books (sorted by rating):
1. 1984 — 4.7
2. Dune — 4.6

Most Expensive Book per Genre:
Thriller: Inferno (450)
SelfHelp: Atomic Habits (550)
ScienceFiction: Dune (600)
History: Sapiens (650)
Fantasy: The Hobbit (480)
Biography: Becoming (700)
Fiction: To Kill a Mockingbird (400)
```