**Python Assignment - 3**

**1. How can you replace string space with a given character in Python (*Note :Do not use string's in built replace function)*?**

**Example 1**: a user has provided the string “S p rat d” and the character “e”, and the output will be “Seperated”.

**Example 2**: a user has provided the string “N ti n” and the character “o”, and the output will be "Notion".

**2.** **Write a python program to identify whether the given number input is an armstrong number or not.** (***Note: Input should be dynamic***)

An Armstrong number is a special kind of number in math. It's a number that equals the sum of its digits, each raised to a power. For example, if you have a number like 153, it's an Armstrong number because 1^3 + 5^3 + 3^3 equals 153.

**3. For a given string. Find the number of occurrence of each letter (should return a dictionary where letters are keys and occurrence is value) and which letter(s) occurred the most.**

**Input**="String Manipulation"

**Output** - dictionary = {"S"=1, "t"=2,"r"=1,"i"=3,"n"=3,"g"=1," "=1,"M"=1,"a"=1,"p"=1,"u"=1,"l"=1,"o"=1}

Letter(s) occurred most are: "i","n"

**4. Write a function to calculate the factorial of a number.**

**Example Input:** 5

**Example Output:** 120

**Hint:** Use a recursion to calculate the factorial.

**5. Write a function that takes a list of numbers and returns the sum of only even numbers.  
Input**: [1, 2, 3, 4, 5, 6]  
**Output**: 12

**6. Write a function that takes two arguments and returns their greatest common divisor (GCD).  
Input**: 48, 18  
**Output**: 6

**7. Write a function that returns the nth Fibonacci number using recursion.  
Input**: fibonacci(5)  
**Output**: 5

**8. Write a function that accepts a list of integers and returns a new list with all duplicate elements removed.**  
**Input**: [1, 2, 2, 3, 4, 4, 5]  
**Output**: [1, 2, 3, 4, 5]

**9. Write a function using lambda to sort a list of tuples by the second element in each tuple.  
Input**: [(1, 'b'), (2, 'a'), (3, 'c')]  
**Output**: [(2, 'a'), (1, 'b'), (3, 'c')]

**10. Write a function that generates all possible permutations of a list of numbers.**  
**Input**: [1, 2, 3]  
**Output**: [[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]]

**11. Write a function that accepts a dictionary and returns the key with the highest value.**  
**Input**: {"a": 10, "b": 15, "c": 5}  
**Output**: "b"

**12. Write a function that flattens a nested list (list of lists) into a single list.**  
**Input**: [[1, 2, 3], [4, 5], [6, 7, 8]]  
**Output**: [1, 2, 3, 4, 5, 6, 7, 8]

**13. Write a function that takes a string as input and returns a dictionary where the keys are the words in the string and the values are the number of times each word appears.**

**Input**: "the quick brown fox jumps over the lazy dog"

**Output**: {'the': 2, 'quick': 1, 'brown': 1, 'fox': 1, 'jumps': 1, 'over': 1, 'lazy': 1, 'dog': 1}

**14. Write a function that takes a list of integers as input and returns the maximum product of two numbers in the list.**

**Input**: [2, 3, -2, 4]

**Output**: 8 (product of 2 and 4)

**15. Write a function that takes a string as input and returns the Roman numeral equivalent.**

**Input**: "12"

**Output**: "XII"