

Practice Exercises - Chapter: 08

* Exercise 8.1: Sum of odd numbers

Write a program computing the sum of odd numbers smaller than n ($1 + 3 + 5 + 7 + \dots$). Input n from the keyboard.

Solution 8.1:

```
# Python Program to Calculate Sum of Odd Numbers from 1 to N

maximum = int(input(" Please Enter the Maximum Value : "))

total = 0

for number in range(1, maximum+1):

    if(number % 2 != 0):

        print("{0}".format(number))

        total = total + number

print("The Sum of Even Numbers from 1 to {0} = {1}".format(number,
total))
```

* Exercise 8.2: Middle string

Write a function taking two strings, s1 and s2 as parameters and returning a new string by appending s2 in the middle of s1. For example, s1 = “abb”, s2 = “ccc”, the result is “accbb”.

Solution 8.2:

```
def append_middle(s1, s2):

    print("Original Strings are", s1, s2)
```

```

# middle index number of s1

mi = int(len(s1) / 2)

# get character from 0 to the middle index number from s1

x = s1[:mi:]

# concatenate s2 to it

x = x + s2

# append remaining character from s1

x = x + s1[mi:]

print("After appending new string in middle:", x)

```

```
append_middle("abb", "ccc")
```

*** Exercise 8.3: Reversing a string**

Write a program reversing a given string.

Solution 8.3:

```

str1 = "PYnative"

print("Original String is:", str1)

str1 = str1[::-1]

print("Reversed String is:", str1)

```

*** Exercise 8.4: Counting characters**

Write a program counting all lower case, upper case, digits, and special symbols from a given string. For example, string “A*45bc%^-gbB1<” has 6 characters, 3 digits, and 5 symbols.

Solution 8.4:

```
def find_digits_chars_symbols(sample_str):  
    char_count = 0  
    digit_count = 0  
    symbol_count = 0  
    for char in sample_str:  
        if char.isalpha():  
            char_count += 1  
        elif char.isdigit():  
            digit_count += 1  
        # if it is not letter or digit then it is special symbol  
        else:  
            symbol_count += 1  
  
    print("Chars =", char_count, "Digits =", digit_count, "Symbol  
=", symbol_count)  
  
sample_str = "A*45bc%^-gbB1<"  
print("total counts of chars, Digits, and symbols \n")  
find_digits_chars_symbols(sample_str)
```

*** Exercise 8.5: Prime number**

Write a function `isPrimeNumber(n)` to check if `n` is a prime number. The function returns True or False. Then, write a program to print all prime numbers that are smaller than 20.

*** Exercise 8.6: Palindrome number**

Write a function to check if the given number is a palindrome number. The function returns True or False. A palindrome number is a number that is same after reverse. For example 98189, is a palindrome number. Write a program using that function.

*** Exercise 8.7: Counting characters occurrences**

Write a program counting occurrences of all characters within a string. For example, the input is "Letter", the output is {'L':1, 'e':2, 't':2, 'r': 1}. Hint: use dictionary.

*** Exercise 8.8: Counting word occurrences**

Write a program counting occurrences of all words within a text. For example, the input is "I chose the red color, because I like red", the output is {"I":2, "chose":1, "the":1, "red":2, "color":1, "because":1, "like":1}. Hint: use dictionary and function split(str) returning a list with each element as a word in str.

*** Exercise 8.9: Matrix multiplication**

Write a function to multiply two matrices: A[n,m] and B[n,o]. Then, check the result by using the Numpy library (function matmul()).

Hint: use nested list to represent matrix. For example, the list [[1,2], [3,4], [4,5]] represents a 3x2 matrix. Use the statement: $X = [[0 \text{ for } i \text{ in range}(m)] \text{ for } j \text{ in range}(n)]$ to create an empty matrix X of n rows and m columns.

*** Exercise 8.10: Evaluating and plotting polynomial**

Write a function to evaluate a polynomial $f(x)$, defined by its coefficients ($c[0] + c[1].x + c[2].x^2 + \dots$), at a given point x. Hint: $x**i$ computes x^i . Then, check the result by using the Numpy library (function polyval()).

Plot the polynomial with $X = [-4, -3, -2, -1, 0, 1, 2, 3, 4]$. Hint: use library Matplotlib.

For example, the graph of $f(x) = 2x^3 - x^2 + 1$ is illustrated in the figure bellow.

