BY Aditya Kundu

SID : 21107003

Subject : Python

Assignment_6

Que_1.

```python
# Takiong integer input from
n = int(input("Enter any number: "))
sum1 = 0   # this will be used to add divisble numbers

# checking if it is perfect number
if n > 0:   # Makeing sure number is positive
    for i in range(1, n):
        if(n % i == 0):
            sum1 = sum1 + i
    if (sum1 == n):
        print("The number is a Perfect number!")
    else:
        print("The number is not a Perfect number!")
else :
    print("Only positve integers can be perfect number.")

```

```
PS D:\Python\Assignment 6> python .\21107003_Assignment_6_Q1.py
Enter any number: 5
The number is not a Perfect number!
PS D:\Python\Assignment 6> python .\21107003_Assignment_6_Q1.py
Enter any number: 28
The number is a Perfect number!
PS D:\Python\Assignment 6> python .\21107003_Assignment_6_Q1.py
Enter any number: 0
Only positve integers can be perfect number.
PS D:\Python\Assignment 6>
```

Que_2.

```python
# function which return reverse of a string

def isPalindrome(s):
    return s == s[::-1]


# taking input
print("Kindly enter alphabets only.")
s = input("Enter word/sentence to be checked: ").replace(" ","").lower()
ans = isPalindrome(s)   # calling function

# checking palindrom
if ans:
    print("Yes")
else:
    print("No")

```

```
PS D:\Python\Assignment 6> python .\21107003_Assignment_6_Q2.py
Kindly enter alphabets only.
Enter word/sentence to be checked: Nun
Yes
PS D:\Python\Assignment 6> python .\21107003_Assignment_6_Q2.py
Kindly enter alphabets only.
Enter word/sentence to be checked: nurses run
Yes
PS D:\Python\Assignment 6>
```

Que_3.

```python
# Print Pascal's Triangle in Python
from math import factorial

r = int(input("Enter the number of rows: "))    # r for rows
for i in range(r):
    for j in range(r-i+1):

        # for left spacing
        print(end=" ")

    for j in range(i+1):

        # nCr = n!/((n-r)!*r!)
        print(factorial(i)//(factorial(j)*factorial(i-j)), end=" ")

    # for new line
    print()
```

```
PS D:\Python\Assignment 6> python .\Assignment_6_Q3.py
Enter the number of rows: 6
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
PS D:\Python\Assignment 6>
```

Que_4.

```python

def pangram(str):
    All_alphabets = "abcdefghijklmnopqrstuvwxyz"
    for i in All_alphabets:
        if i not in str.lower():   # As python is case sensitive, so we have to check its small letters too
            return False

    return True

# Example:
string = input("Enter the sentence: ")
if(pangram(string) == True):
    print("Yes, The given string is pangram.")
else:
    print("No, the given string is not a pangram.")

```

```
PS D:\python\Assignment 6> python .\21107003_Assignment_6_Q4.py
Enter the sentence: The quick brown fox jumps over the lazy dog
Yes, The given string is pangram.
PS D:\python\Assignment 6> python .\21107003_Assignment_6_Q4.py
Enter the sentence: abcd
No, the given string is not a pangram.
PS D:\python\Assignment 6>
```

Que_5.

```python
# Creating list and input words separated by '-' as elements.
items=[n for n in input("Enter sentence here: ").split('-')]


items.sort()    # arranging alphabetically

print('-'.join(items))

```

```
PS D:\python\Assignment 6> python .\21107003_Assignment_6_Q5.py
Enter sentence here: green-red-yellow-black-white
black-green-red-white-yellow
PS D:\python\Assignment 6>
```

Que_6.

```python
# using dictionary argument to print out student detail according to data given.
def student_data(student_id, **kwargs):
    print(f'\nStudent ID: {student_id}')
    if 'student_name' in kwargs:
        print(f"Student Name: {kwargs['student_name']}")

    if 'student_name' and 'student_class' in kwargs:
            print(f"\nStudent Name: {kwargs['student_name']}")
            print(f"Student Class: {kwargs['student_class']}")


# Calling function and giving only student name
student_data(student_id='21107003', student_name='Aditya Kundu')

# Calling function and giving both student name and class
student_data(student_id='21107005', student_name='Harshvardhan', student_class ='21-25')

```

```
PS D:\python\Assignment 6> python .\21107003_Assignment_6_Q6.py

Student ID: 21107003
Student Name: Aditya Kundu

Student ID: 21107005
Student Name: Harshvardhan

Student Name: Harshvardhan
Student Class: 21-25
PS D:\python\Assignment 6>
```

Que_7.

```python
# creating class
class Student:
    pass
class Marks:
    pass
# instance
student1 = Student()
marks1 = Marks()
#Checking whether the instance or not.
print(isinstance(student1, Student))
print(isinstance(marks1, Student))
print(isinstance(marks1, Marks))
print(isinstance(student1, Marks))
print("\nChecking whether the said classes are subclasses of the built-in object class or not....")
print(issubclass(Student, object))
print(issubclass(Marks, object))
```

```
PS D:\python\Assignment 6> python .\21107003_Assignment_6_Q7.py
True
False
True
False

Checking whether the said classes are subclasses of the built-in object class or not....
True
True
PS D:\python\Assignment 6>
```

Que_8.

```python

def three_Sum(num):
    if len(num)<3: return []
    num.sort()
    result=[]
    for i in range(len(num)-2):
        left=i+1
        right=len(num)-1
        if i!=0 and num[i]==num[i-1]:continue
        while left<right:
            if num[left]+num[right]==-num[i]:
                result.append([num[i],num[left],num[right]])
                left=left+1
                right=right-1
                while num[left]==num[left-1] and left<right:left=left+1
                while num[right]==num[right+1] and left<right: right=right-1
            elif num[left]+num[right]<-num[i]:
                left=left+1
            else:
                right=right-1
    return result

nums1=[-25, -10, -7, -3, 2, 4, 8, 10]

print(three_Sum(nums1))

```

```
PS D:\python\Assignment 6> python .\21107003_Assignment_6_Q8.py
[[-10, 2, 8], [-7, -3, 10]]
PS D:\python\Assignment 6>
```

Que_9.

```python

class validation:
    def is_valid_parenthese(self, str1):
        #Here two datatypes are used such as list and dictionary providing a key value pairs.
        stack, pchar = [], {"(": ")", "{": "}", "[": "]"}
        #Using for loop we will check the desired keys and add them in stack.
        for parenthese in str1:
            if parenthese in pchar:
                stack.append(parenthese)
            #If not in pchar, we will pop that parantheses!!!
            elif len(stack) == 0 or pchar[stack.pop()] != parenthese:
                return False
        return len(stack) == 0

# we call the function here!
print(validation().is_valid_parenthese("(){}[]"))
print(validation().is_valid_parenthese("()"))
print(validation().is_valid_parenthese("[)"))
print(validation().is_valid_parenthese("({[)]"))
print(validation().is_valid_parenthese("{{{"))

```

```
PS D:\python\Assignment 6> python .\21107003_Assignment_6_Q9.py
True
True
False
False
False
PS D:\python\Assignment 6>
```