```python
#Q1          SID-21107052
#Python program for checking if input number is perfect or not
def perfect(num):
    sum=0
    for i in range(1,num):
        if num%i==0:
            sum +=i
    if sum==num:
        print("it is perfect")
    else:
        print("it is not perfect")
num=int(input("enter no. to be checked if perfect: "))
perfect(num)
```

```
enter no. to be checked if perfect: 496
it is perfect


...Program finished with exit code 0
Press ENTER to exit console.
```

```python
#Q2          SID-21107052
#Python program for checking if input word is palindrome or not
def palindrome(word):
    for i in range(0,len(word)):
        if word[i]!=word[~i]:
            print("it is not a palindrome")
            break
    else:
        print("it is a palindrome")
word=input("enter word to be checked: ")
palindrome(word)
```

```
enter word to be checked: racecar
it is a palindrome


...Program finished with exit code 0
Press ENTER to exit console.
```

```python
#Q3          SID-21107052
#Python program for printing rows of pascal's triangle
from math import factorial

n=int(input("enter no. of rows to be printed: "))
for i in range(n):
    for j in range(n-1-i):
        print(" ", end="")
    for k in range (i+1):
        #we use combination formula ie. nCr
        print(factorial(i) // (factorial(i-k)*factorial(k)) , end=" ")
    print()
```

```
enter no. of rows to be printed: 7
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
1 6 15 20 15 6 1


...Program finished with exit code 0
Press ENTER to exit console.
```

```python
#Q4          SID-21107052
#Python program for checking if input sentence is panagram or not
alphabet='abcdefghijklmnopqrstuvwxyz'
def panagram(sentence):
    for i in alphabet:
        if i not in sentence:
            print("it is not a panagram")
            break
    else:
        print("it is panagram")
sentence=input("enter the sentence to be check: ")
panagram(sentence)
```

```
enter the sentence to be check: the quick brown fox jumps overs a lazy dog
it is panagram


...Program finished with exit code 0
Press ENTER to exit console.
```

```python
#Q5          SID-21107052
#Python program for arranging words alphabetically
def sorter(words):
    words.sort()
    print(*words)
words=list(map(str,input("enter words to be sorted: ").split('-')))
sorter(words)
```

```
enter words to be sorted: green-black-red-orange
black green orange red


...Program finished with exit code 0
Press ENTER to exit console.
```

```python
#Q6          SID-21107052
#Python program for printing id of student
def student_data(student_id, **kwargs):
    print('\nStudent ID: ',student_id)
    if 'student_name' in kwargs:
        print("Student Name: ", kwargs['student_name'])

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

student_data(student_id='21107052', student_name=' Gurkirat Singh')

student_data(student_id='21107052', student_name='Gurkirat Singh', student_class ='Mechanical')
```

input

```
Student ID:  21107052
Student Name:   Gurkirat Singh

Student ID:  21107052
Student Name:  Gurkirat Singh

Student Name:  Gurkirat Singh
Student Class:  Mechanical


...Program finished with exit code 0
Press ENTER to exit console.
```

```python
#Q7          SID-21107052
#Python program for creating classes and check if given instances are part of these classes or not
class Student:
    pass
class Marks:
    pass
student_ = Student()
marks_ = Marks()
print(isinstance(student_, Student))
print(isinstance(marks_, Student))
print(isinstance(marks_, Marks))
print(isinstance(student_, Marks))
print("\nCheck whether the said classes are subclasses of the built-in object class or not.")
print(issubclass(Student, object))
print(issubclass(Marks, object))
```

input

```
True
False
True
False

Check whether the said classes are subclasses of the built-in object class or not.
True
True


...Program finished with exit code 0
Press ENTER to exit console.
```

```python
#Q8          SID-21107052
#Python program for finding 3 no.s in given list if their sum is 0
class py_solution:
 def Sum(self, nums):
        nums, result, i = sorted(nums), [], 0
        while i < len(nums) - 2:
            j, k = i + 1, len(nums) - 1
            while j < k:
                if nums[i] + nums[j] + nums[k] < 0:
                    j += 1
                elif nums[i] + nums[j] + nums[k] > 0:
                    k -= 1
                else:
                    result.append([nums[i], nums[j], nums[k]])
                    j, k = j + 1, k - 1
                    while j < k and nums[j] == nums[j - 1]:
                        j += 1
                    while j < k and nums[k] == nums[k + 1]:
                        k -= 1
            i += 1
            while i < len(nums) - 2 and nums[i] == nums[i - 1]:
                i += 1
        return result

print(py_solution().Sum([-25, -10, -7, -3, 2, 4, 8, 10]))
```

```
[[-10, 2, 8], [-7, -3, 10]]


...Program finished with exit code 0
Press ENTER to exit console.
```

```python
#Q9          SID-21107052
#Python program to check if input brackets are closed or not
class parantheses:
    def find(str):
        a= ['()', '{}', '[]']
        while any(i in str for i in a):
            for j in a:
                str = str.replace(j, '')
        return not str

s = input("Enter the sequence of parantheses : ")
if parantheses.find(s):
    print(s,"is balanced",sep=" - ")
else:
    print(s,"is unbalanced",sep=" - ")
```

```
Enter the sequence of parantheses : {{}}[]
{{}}[] - is balanced


...Program finished with exit code 0
Press ENTER to exit console.
```