ASSIGNMENT

NAME- RAJAT SINGH

ROLL NO.- 46

UNIV. ROLL NO.- 2215001414

SECTION- U

SUBJECT- PYTHON PROGRAMMING

1. Python Program for n-th Fibonacci number.

ANS-

```
def fibonacci(n):
    if n <= 1:
        return n
    else:
        return fibonacci(n-1) + fibonacci(n-2)

n = int(input("Enter a positive integer: "))

if n < 0:
    print("Please enter a positive integer.")
else:
    print("The", n, "th Fibonacci number is", fibonacci(n))
```

2. Python Program for How to check if a given number is Fibonacci number?

ANS-

```
def is_fibonacci(n):
    if n < 0:
        return False
    elif n == 0 or n == 1:
        return True
    else:
        a, b = 0, 1
        while b < n:
            a, b = b, a + b
        return b == n
```

3. Python Program for n\'th multiple of a number in Fibonacci Series

ANS-

```
def find_nth_multiple(n, k):
    a, b = 0, 1
    multiple_count = 0

    while True:
        if a % k == 0:
            multiple_count += 1
            if multiple_count == n:
                return a

        a, b = b, a + b
```

```
n = int(input("Enter the value of n: "))
k = int(input("Enter the value of k: "))

if k == 0:
    print("Please enter a non-zero value for k.")
else:
    nth_multiple = find_nth_multiple(n, k)
    print("The", n, "th multiple of", k, "in the Fibonacci series is", nth_multiple)
```

4. Program to print ASCII Value of a character

ANS-

```
char = input("Enter a character: ")
ascii_value = ord(char)
print("The ASCII value of", char, "is", ascii_value)
```

5. Python Program for Sum of squares of first n natural numbers

ANS-

```
n = int(input("Enter the value of n: "))
sum_of_squares = 0
for i in range(1, n+1):
    sum_of_squares += i*i
print("The sum of squares of first", n, "natural numbers is:", sum_of_squares)
```

6. Write a Python program to swap two numbers using bitwise operator.

ANS-

```
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))
print("Original values:")
print("a =", a)
print("b =", b)
a = a ^ b
b = a ^ b
a = a ^ b
print("Swapped values:")
print("a =", a)
print("b =", b)
```

7. Write a Python program to check whether a character is alphabet or not.

ANS-

```
character = input("Enter a character: ")
if character.isalpha():
    print("The character is an alphabet")
else:
    print("The character is not an alphabet")
```

8. Write a Python program to input any alphabet and check whether it is vowel or consonant.

ANS-

```
alphabet = input("Enter an alphabet: ")
alphabet = alphabet.lower()
if alphabet in ('a', 'e', 'i', 'o', 'u'):
    print(f"{alphabet} is a vowel.")
else:
    print(f"{alphabet} is a consonant.")
```

9. Write a Python program to input any character and check whether it is alphabet, digit
or special character.

ANS-

```
char = input("Enter a character: ")
if char.isalpha():
    print(f"{char} is an alphabet.")
elif char.isdigit():
    print(f"{char} is a digit.")
else:
    print(f"{char} is a special character.")
```

10. Write a Python program to input marks of five subjects Physics, Chemistry, Biology, Mathematics and Computer. Calculate percentage and grade according to following:

Percentage >= 90% : Grade A

Percentage >= 80% : Grade B

Percentage >= 70% : Grade C

Percentage >= 60% : Grade D

Percentage >= 40% : Grade E

Percentage < 40% : Grade F

ANS-

```
physics = float(input("Enter marks in Physics: "))
chemistry = float(input("Enter marks in Chemistry: "))
biology = float(input("Enter marks in Biology: "))
mathematics = float(input("Enter marks in Mathematics: "))
computer = float(input("Enter marks in Computer: "))
total_marks = physics + chemistry + biology + mathematics + computer
percentage = (total_marks / 500) * 100
if percentage >= 90:
    grade = "A"
elif percentage >= 80:
    grade = "B"
elif percentage >= 70:
    grade = "C"
elif percentage >= 60:
```

```
    grade = "D"
elif percentage >= 40:
    grade = "E"
else:
    grade = "F"
print(f"Percentage: {percentage:.2f}%")
print(f"Grade: {grade}")
```

11. Write a Python program to input basic salary of an employee and calculate its Gross salary according to following:
    Basic Salary <= 10000 : HRA = 20%, DA = 80%
    Basic Salary <= 20000 : HRA = 25%, DA = 90%
    Basic Salary > 20000 : HRA = 30%, DA = 95%

ANS-

```
basic_salary = float(input("Enter the basic salary of the employee: "))
if basic_salary <= 10000:
    hra = basic_salary * 0.2
    da = basic_salary * 0.8
elif basic_salary <= 20000:
    hra = basic_salary * 0.25
    da = basic_salary * 0.9
else:
    hra = basic_salary * 0.3
    da = basic_salary * 0.95
gross_salary = basic_salary + hra + da
print(f"Gross salary: {gross_salary:.2f}")
```

12. Write a Python program to input electricity unit charges and calculate total electricity
bill according to the given condition:
For first 50 units Rs. 0.50/unit
For next 100 units Rs. 0.75/unit
For next 100 units Rs. 1.20/unit
For unit above 250 Rs. 1.50/unit
An additional surcharge of 20% is added to the bill

ANS-

```
units = float(input("Enter the electricity units consumed: "))
if units <= 50:
    total_bill = units * 0.5
elif units <= 150:
    total_bill = 25 + (units - 50) * 0.75
elif units <= 250:
    total_bill = 100 + (units - 150) * 1.20
else:
    total_bill = 220 + (units - 250) * 1.50
total_bill *= 1.20
print(f"Total electricity bill: Rs. {total_bill:.2f}")
```

13. Write a Python program to print all alphabets from a to z. – using while Loop

ANS-

```
char = 'a'
while char <= 'z':
    print(char)
    char = chr(ord(char) + 1)
```

14. Write a Python program to find first and last digit of a number.

ANS-

```
num = int(input("Enter a number: "))
first_digit = num
while first_digit >= 10:
    first_digit //= 10
last_digit = num % 10
print("First digit:", first_digit)
print("Last digit:", last_digit)
```

15. Write a Python program to calculate sum of digits of a number.

ANS-

```
num = int(input("Enter a number: "))
sum = 0
while num > 0:
    digit = num % 10
    sum += digit
```

```
    num //= 10
print("Sum of digits:", sum)
```

16. Write a Python program to calculate product of digits of a number.

ANS-

```
num = int(input("Enter a number: "))
product = 1
while num > 0:
    digit = num % 10
    product *= digit
    num //= 10
print("Product of digits:", product)
```

17. Write a Python program to enter a number and print its reverse.

ANS-

```
num = int(input("Enter a number: "))
reverse = 0
temp = num
while temp > 0:
    digit = temp % 10
    reverse = reverse * 10 + digit
    temp //= 10
print("Reverse of the number:", reverse)
```

18. Write a Python program to check whether a number is palindrome or not.

ANS-

```
num = int(input("Enter a number: "))
reverse = 0
temp = num
while temp > 0:
    digit = temp % 10
    reverse = reverse * 10 + digit
    temp //= 10
if num == reverse:
    print(num, "is a palindrome")
else:
    print(num, "is not a palindrome")
```

19. Write a Python program to find all factors of a number.

ANS-

```
num = int(input("Enter a number: "))
factors = []
for i in range(1, num+1):
    if num % i == 0:
        factors.append(i)
print("Factors of", num, "are:", factors)
```

20. Write a Python program to calculate factorial of a number

ANS-

```
num = int(input("Enter a number: "))
factorial = 1
for i in range(1, num+1):
    factorial *= i
print("Factorial of", num, "is", factorial)
```

21. Write a Python program to find HCF (GCD) of two numbers.

ANS-

```
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
if num1 < num2:
    smaller = num1
else:
    smaller = num2
for i in range(smaller, 0, -1):
    if num1 % i == 0 and num2 % i == 0:
        hcf = i
        break
print("HCF of", num1, "and", num2, "is", hcf)
```

22. Write a Python program to find LCM of two numbers.

ANS-

```
num1 = int(input("Enter the first number: "))
num2 = int(input("Enter the second number: "))
if num1 > num2:
   maximum = num1
else:
   maximum = num2
while True:
   if maximum % num1 == 0 and maximum % num2 == 0:
      lcm = maximum
      break
   maximum += 1
print("LCM of", num1, "and", num2, "is", lcm)
```

23. Write a Python program to check whether a number is Prime number or not.

ANS-

```
num = int(input("Enter a number: "))
if num > 1:
   for i in range(2, int(num ** 0.5) + 1):
      if num % i == 0:
         print(num, "is not a prime number")
         break
   else:
      print(num, "is a prime number")
else:
```

```
    print(num, "is not a prime number")
```

24. Write a Python program to print all Prime numbers between 1 to n

ANS-

```
n = int(input("Enter a number: "))
for num in range(1, n + 1):
    if num > 1:
        for i in range(2, int(num ** 0.5) + 1):
            if num % i == 0:
                break
        else:
            print(num)
```

25. Write a Python program to find sum of all prime numbers between 1 to n.

ANS-

```
n = int(input("Enter a number: "))
sum_of_primes = 0
for num in range(1, n + 1):
    if num > 1:
        for i in range(2, int(num ** 0.5) + 1):
            if num % i == 0:
                break
        else:
            sum_of_primes += num
print("The sum of all prime numbers between 1 and", n, "is", sum_of_primes)
```

26. Write a Python program to find all prime factors of a number

ANS-

```
def prime_factors(n):
    factors = []
    while n % 2 == 0:
        factors.append(2)
        n = n // 2
    for i in range(3, int(n**0.5)+1, 2):
        while n % i == 0:
            factors.append(i)
            n = n // i
    if n > 2:
        factors.append(n)
    return factors
```

27. Write a Python program to check whether a number is Armstrong number or not.

ANS-

```
def is_armstrong(n):
    num_str = str(n)
    length = len(num_str)
    sum = 0
    for digit in num_str:
        sum += int(digit)**length
    return sum == n
```

28. Write a Python program to print all Armstrong numbers between 1 to n

ANS-

```
def is_armstrong(n):
    num_str = str(n)
    length = len(num_str)
    sum = 0
    for digit in num_str:
        sum += int(digit)**length
    return sum == n
def armstrong_numbers(n):
    for i in range(1, n+1):
        if is_armstrong(i):
            print(i)
```

29. Write a Python program to check whether a number is Perfect number or not.

ANS-

```
def is_perfect(n):
    sum = 0
    for i in range(1, n//2+1):
        if n % i == 0:
            sum += i
    return sum == n
```

30. Write a Python program to check whether a number is Strong number or not (Also known as Robinson number/ Krishnamurthy Number / Peterson number.)

ANS-

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
def is_strong(n):
    num_str = str(n)
    sum = 0
    for digit in num_str:
        sum += factorial(int(digit))
    return sum == n
```

31. Python program to check whether the string is Symmetrical or Palindrome

ANS-

```
def is_symmetrical(s):
    return s == s[::-1]
def is_palindrome(s):
    s = ''.join(filter(str.isalnum, s))
    s = s.lower()
    return is_symmetrical(s)
```

32. Reverse words in a given String in Python

ANS-

```
def reverse_words(s):
    words = s.split()
    words.reverse()
    return ' '.join(words)
```

33. Ways to remove i’th character from string in Python

ANS-

There are several ways to remove I th character from a string

1 String slicing

2 String concatenation

3 List comprehension

STRING SLICING

```
def remove_char(s, i):
    return s[:i] + s[i+1:]
```

34. Python program to Check if a Substring is Present in a Given String

Ans-

```
def is_substring(s, sub):
    if sub in s:
        return True
    else:
        return False
```

35. Python program to count words frequency in String Shorthands

Ans-

```
from collections import Counter

s = "This is a test string. This string is just a test."

words = s.split()

word_counts = Counter(words)

print(word_counts)
```

36. Python program to convert snake case to pascal case

Ans-

```
def snake_to_pascal_case(s):
    words = s.split('_')
    pascal_case = ''.join(word.capitalize() for word in words)
    return pascal_case
```

37. Find length of a string in python (4 ways)

Ans-

Four ways to find the length of a string in python-

1. Using the `len()` function:
2. Using a loop:
3. Using the `sum()` function with a generator expression:
4. Using recursion:

38. Python program to print even length words in a string

Ans-

```
def print_even_length_words(s):
    words = s.split()
    for word in words:
        if len(word) % 2 == 0:
            print(word)
```

39. Python program to accept the strings which contains all vowels

Ans-

```
def contains_all_vowels(s):
    vowels = {'a', 'e', 'i', 'o', 'u'}
    return vowels.issubset(set(s.lower()))
```

40. Python program to count the Number of matching characters in a pair of string

Ans-

```
def count_matching_chars(s1, s2):
    count = 0
    for c1, c2 in zip(s1, s2):
        if c1 == c2:
            count += 1
    return count
```

41. Remove all duplicates from a given string in Python

Ans-

```
def remove_duplicates(s):
    return ''.join(sorted(set(s), key=s.index))
```

42. Python programs to count Least Frequent Character in String

Ans-

```
def count_least_frequent_char(s):
    freq = {}
    for c in s:
        freq[c] = freq.get(c, 0) + 1
    least_freq_char = min(freq, key=freq.get)
    least_freq_count = freq[least_freq_char]
    return least_freq_char, least_freq_count
```

43. Python programs to count maximum frequency character in String

Ans-

```
def count_max_frequency_char(s):
    freq = {}
    for c in s:
        freq[c] = freq.get(c, 0) + 1
    max_freq_char = max(freq, key=freq.get)
    max_freq_count = freq[max_freq_char]
    return max_freq_char, max_freq_count
```

44. Python program to check if a string contains any special character

Ans-

```
import re
def contains_special_char(s):
    regex = re.compile('[@_!#$%^&*()<>?/\|}{~:]')
    if regex.search(s):
        return True
    else:
        return False
```

45. Python program to split and join a string

Ans-

```
s = "The quick brown fox jumps over the lazy dog"
words = s.split()
s_new = " ".join(words)
print("Original string:", s)
print("New string:", s_new)
```

46. Python program to find uncommon words from two Strings

Ans-

```
def uncommon_words(s1, s2):
    words1 = set(s1.split())
    words2 = set(s2.split())
    uncommon = words1.symmetric_difference(words2)
    return list(uncommon)
```

47. Python program to replace duplicate occurrence in string

Ans-

```
def replace_duplicate(s):
    chars = list(s)
    seen = set()
    for i in range(len(chars)):
        if chars[i] in seen:
            chars[i] = '*'
        else:
            seen.add(chars[i])
    new_s = ''.join(chars)
    return new_s
```

48. String slicing in Python to rotate a string

Ans-

```
def rotate_string(s, n):
    split_index = len(s) - n
    rotated_s = s[split_index:] + s[:split_index]
    return rotated_s
```

49.Find all duplicate characters in string

Ans-

```
def find_duplicate_characters(s):
    char_freq = {}
    for char in s:
        if char in char_freq:
            char_freq[char] += 1
        else:
            char_freq[char] = 1
    duplicates = [char for char in char_freq if char_freq[char] > 1]
    return duplicates
```

50.Replace all occurrences of a substring in a string

Ans-

```
def replace_substring(s, old_substring, new_substring):
    new_string = s.replace(old_substring, new_substring)
    return new_string
```