Python Assignment

Name - Deepak Jadon
Roll - 21
Uni. Roll- 2215000544
Course - B.Tech CS
Sec - O

Ans 1:
```
n = int(input("Enter a number: "))
    a, b = 0, 1
    for i in range(n):
        a, b = b, a + b
    print(f"The {n}-th Fibonacci number is {a}.")
```

Ans 2:
```
n = int(input("Enter a number: "))
    a, b = 0, 1
    fib = 0
    while fib < n:
        fib = a + b
        a, b = b, fib
        if fib == n:
            print(f"{n} is a Fibonacci number.")
            break
    else:
        print(f"{n} is not a Fibonacci number.")
```

Ans 3:
```
n = int(input("Enter the value of n: "))
    m = int(input("Enter the value of m: "))
```

```
a, b = 0, 1
fib = 0
while n > 0:
    fib = a + b
    a, b = b, fib
    if fib % m == 0:
        n -= 1
print(f"The {n+1}-th multiple of {m} in the Fibonacci series is {fib}.")
```

Ans 4:

```
alpa='p'
print(ord(alpa))
```

Ans 5:

```
n=int(input("Enter the number"))
Sum=0
for i in range(1,n+1):
    Sum=sum+(i*i)
print(sum)
```

Ans 6:

```
a=int(input())
b=int(input())
a,b=b,a
print(a,b)
```

Ans 7: n=intput()

```
if(n>='A' and n<='Z' or n>='a' and n<='z'):
        print("It is a alphabet")
else:
        print("It is not a alphabet")
```

Ans 8: n=intput()

```
if(n=='a' or n=='e' or n=='i' or n=='o' or n=='u' or n=='A' or n=='E' or n=='I' or n=='O' or n=='U'):
        print("It is a vowel")
else:
        print("It is a consonent")
```

Ans 9: ch=input()

```
if((ch>='a'  and ch<='z') or (ch>='A' and ch<='Z')):
        print("The given Character is a alphabet")
elif(ch>='0' and ch<='9'):
        print("The given character is a digit")
else:
        print("The given character is a special character")
```

Ans 10: sub1=int(input())

```
  sub2=int(input())
```

```
sub3=int(input())
sub4=int(input())
sub5=int(input())
percentage=(sub1+ sub2+ sub3+ sub4+ sub5)/5
if(percentage>=90):
        print("Grade=A")
elif(percentage>=80):
        print("Grade=B")
elif(percentage>=70):
        print("Grade=C")
elif(percentage>=60):
        print("Grade=D")
elif(percentage>=40):
        print("Grade=E")
else:
        print("Grade=F")
```

Ans 11:

```
basic_salary=int(input("Enter your basic salary"))
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("Your gross salary is ",gross_salary)
```

Ans 12:

```
units = int(input("Enter the number of units consumed: "))
total_bill = 0
surcharge_percent = 20
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
surcharge_amount = (surcharge_percent / 100) * total_bill
```

```
        total_bill += surcharge_amount
        print(f"Total electricity bill (including {surcharge_percent}% surcharge): Rs. {total_bill:.2f}")
```

Ans 13:

```
start = ord('a')
end = ord('z')
while start <= end:
        print(chr(start), end=' ')
        start += 1
```

Ans 14:

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

Ans 15:

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

```
        sum_of_digits += digit
        temp //= 10
    print(f"Sum of digits of {num}: {sum_of_digits}")
```

Ans 16:

```
num = int(input("Enter a number: "))
product_of_digits = 1
temp = num
while temp > 0:
    digit = temp % 10
    product_of_digits *= digit
    temp //= 10
print(f"Product of digits of {num}: {product_of_digits}")
```

Ans 17:

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

Ans 18:

```
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(f"{num} is a palindrome number")
else:
    print(f"{num} is not a palindrome number")
```

Ans 19:

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

Ans 20:

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

```
print(f"The factorial of {num} is: {factorial}")
```

Ans 21:

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

Ans 22:

```
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))
if num1 > num2:
    greater = num1
else:
    greater = num2
lcm = greater
while True:
```

```
            if lcm % num1 == 0 and lcm % num2 == 0:
                break
            lcm += greater
        print(f"The LCM of {num1} and {num2} is: {lcm}")
```

Ans 23:

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

Ans 24:

```
n = int(input("Enter the upper limit: "))
        for num in range(2, n+1):
            if num > 1:
                for i in range(2, num):
                    if num % i == 0:
                        break
```

```
            else:
                print(num)
```

Ans 25:

```
n = int(input("Enter the upper limit: "))
prime_sum = 0
for num in range(2, n+1):
    if num > 1:
        for i in range(2, num):
            if num % i == 0:
                break
        else:
            prime_sum += num
print(f"The sum of all prime numbers between 1 and {n} is {prime_sum}.")
```

Ans 26:

```
num = int(input("Enter a number: "))
prime_factors = []
for i in range(2, num+1):
    while num % i == 0:
        prime_factors.append(i)
        num //= i
if num == 1:
```

```
            break
        print(f"The prime factors of {num} are: {prime_factors}")
```

Ans 27:

```
n=int(input("Enter a number"))
s=n
b=len(str(n))
sum=0
while(n>0):
    r=n%10
    sum=sum+(r**b)
    n=n//10
if s==sum:
    print("The given number",s,"is a armstrong number")
else:
    print("The given number",s,"is not a armstrong number")
```

Ans 28:

```
n = int(input("Enter the upper limit: "))
for num in range(1, n+1):
order = len(str(num))
sum = 0
temp = num
while temp > 0:
```

```
        digit = temp % 10
        sum += digit ** order
        temp //= 10
    if num == sum:
        print(num)
```

Ans 29:

```
num = int(input("Enter a number: "))
sum = 0
for i in range(1, num//2 + 1):
    if num % i == 0:
        sum += i
if num == sum:
    print(num, "is a Perfect number")
else:
    print(num, "is not a Perfect number")
```

Ans 30:

```
num = int(input("Enter a number: "))
sum = 0
temp = num
while temp > 0:
    digit = temp % 10
    fact = 1
```

```
        for i in range(1, digit+1):
            fact *= i
        sum += fact
        temp //= 10
if num == sum:
    print(num, "is a Strong number")
else:
    print(num, "is not a Strong number")
```

Ans 31:

```
string = input("Enter a string: ")
if string == string[::-1]:
    print("The string is symmetrical")
else:
    print("The string is not symmetrical")
if string.replace(" ", "") == string.replace(" ", "")[::-1]:
    print("The string is a palindrome")
else:
    print("The string is not a palindrome")
```

Ans 32:

```
string = input("Enter a string: ")
words = string.split()
words.reverse()
```

```
new_string = " ".join(words)
print("The string with reversed words is:", new_string)
```

Ans 33: (a) Using slicing:-

```
string = input("Enter a string: ")
i = int(input("Enter the index of the character to remove: "))
new_string = string[:i] + string[i+1:]
print("The new string with the i'th character removed is:", new_string)
```

(b) Using String concatination:-

```
string = input("Enter a string: ")
i = int(input("Enter the index of the character to remove:"))
new_string = ""
for j in range(len(string)):
    if j != i:
        new_string += string[j]
print("The new string with the i'th character removed is:", new_string)
```

(c) Using list comprehenion and join:-

```
string = input("Enter a string: ")
i = int(input("Enter the index of the character to remove: "))
```

```
new_string = "".join([string[j] for j in range(len(string)) if j != i])
print("The new string with the i'th character removed is:", new_string)
```

Ans 34:
```
string = input("Enter a string: ")
substring = input("Enter a substring: ")
if substring in string:
    print(f"{substring} is present in {string}.")
else:
    print(f"{substring} is not present in {string}.")
```

Ans 35:
```
string = input("Enter a string: ")
shorthands = {"u": "you", "r": "are", "2": "to", "4": "for", "b": "be", "n": "and"}
for shorthand, full_form in shorthands.items():
    string = string.replace(shorthand, full_form)
word_freq = {word: string.split().count(word) for word in set(string.split())}
for word, freq in word_freq.items():
    print(f"{word}: {freq}")
```

Ans 36:
```
snake_str = "hello_world"
components = snake_str.split('_')
pascal_case = ''.join(x.title() for x in components)
```

```
print(pascal_case)
```

Ans 37: (a) Using len() function:-

```
string = "Hello, world!"
length = len(string)
print(length)
```

(b) Using a loop:-

```
string = "Hello, world!"
count = 0
for char in string:
    count += 1
print(count)
```

(c) Using len() with a list comprehension:-

```
string = "Hello, world!"
char_list = [char for char in string]
length = len(char_list)
print(length)
```

(d) Using len() function with split() function:-

```
string = "Hello, world!"
word_list = string.split()
length = len(word_list)
```

```
print(length) # Output: 2
```

Ans 38:

```
string = "The quick brown fox jumps over the lazy dog"
words = string.split()
for word in words:
if len(word) % 2 == 0:
    print(word)
```

Ans 39:

```
input_string = input("Enter a string: ")
vowels = set('aeiou')
if vowels.issubset(set(input_string.lower())):
    print("String contains all vowels.")
else:
    print("String does not contain all vowels.")
```

Ans 40:

```
string1 = input("Enter the first string: ")
string2 = input("Enter the second string: ")
matching_chars = 0
for char1 in string1:
    for char2 in string2:
        if char1 == char2:
            matching_chars += 1
print(f"Number of matching characters: {matching_chars}")
```

Ans 41:
```
input_string = input("Enter a string: ")
unique_chars = set(input_string)
result_string = ''.join(unique_chars)
print("String after removing duplicates:", result_string)
```

Ans 42:
```
input_string = input("Enter a string: ")
char_count = {}
for char in input_string:
    if char in char_count:
        char_count[char] += 1
    else:
        char_count[char] = 1
least_frequent_char = None
least_frequent_count = float('inf')
for char, count in char_count.items():
    if count < least_frequent_count:
        least_frequent_char = char
        least_frequent_count = count
print(f"Least frequent character: '{least_frequent_char}' ({least_frequent_count} occurrences)")
```

Ans 43:
```
input_string = input("Enter a string: ")
```

```
char_count = {}
for char in input_string:
    if char in char_count:
        char_count[char] += 1
    else:
        char_count[char] = 1
max_frequency_char = None
max_frequency = 0
for char, count in char_count.items():
    if count > max_frequency:
        max_frequency_char = char
        max_frequency = count
print(f"Character with maximum frequency: '{max_frequency_char}' ({max_frequency} occurrences)")
```

Ans 44:

```
import re
def has_special_char(s):
    pattern = re.compile(r'\W')
    return bool(pattern.search(s))
input_string = input("Enter a string: ")
if has_special_char(input_string):
```

```
        print("The string contains a special character.")
    else:
        print("The string does not contain any special character.")
```

Ans 45:

```
input_string = input("Enter a string: ")
delimiter = input("Enter a delimiter: ")
substring_list = input_string.split(delimiter)
output_string = delimiter.join(substring_list)
print("Original string:", input_string)
print("Substring list:", substring_list)
print("Joined string:", output_string)
```

Ans 46:

```
string1 = input("Enter the first string: ")
string2 = input("Enter the second string: ")
words1 = string1.split()
words2 = string2.split()
all_words = set(words1 + words2)
uncommon_words = [word for word in all_words if (word in words1) ^ (word in words2)]
print("Uncommon words:", uncommon_words)
```

Ans 47:

```
input_string = input("Enter a string: ")
output_string = ""
```

```
for i in range(len(input_string)):
    if input_string[i] not in input_string[:i]:
        output_string += input_string[i]
print("Original string:", input_string)
print("Modified string:", output_string)
```

Ans 48:
```
input_string = input("Enter a string: ")
n = int(input("Enter the number of positions to rotate the string: "))
part1 = input_string[n:]
part2 = input_string[:n]
rotated_string = part1 + part2
print("Original string:", input_string)
print("Rotated string:", rotated_string)
```

Ans 49:
```
input_string = input("Enter a string: ")
duplicates = []
for char in input_string:
    if input_string.count(char) > 1:
        if char not in duplicates:
            duplicates.append(char)
print("Original string:", input_string)
```

```
print("Duplicate characters:", duplicates)
```

Ans 50:

```
input_string = input("Enter a string: ")
substring = input("Enter the substring to replace: ")
replacement = input("Enter the replacement string: ")
new_string = input_string.replace(substring, replacement)

print("Original string:", input_string)
print("New string:", new_string)
```