

Name: Avni tiwari
Section: U
Class roll no.: 11

#Q. 1

```
def Fibonacci(n):  
    if n <= 0:  
        print("Incorrect input")  
    elif n == 1:  
        return 0  
    elif n == 2:  
        return 1  
    else:  
        return Fibonacci(n-1)+Fibonacci(n-2)  
  
print(Fibonacci(10))
```

Q 2

```
n=int(input("Enter the number: "))  
c=0  
a=1  
b=1  
if n==0 or n==1:  
    print("Yes")  
else:  
    while c<n:  
        c=a+b  
        b=a  
        a=c  
    if c==n:  
        print("Yes")  
    else:  
        print("No")
```

Q 3

```
def findPosition(k, n):  
    f1 = 0  
    f2 = 1  
    i = 2;  
    while i!=0:  
        f3 = f1 + f2;  
        f1 = f2;  
        f2 = f3;
```

```

        if f2%k == 0:
            return n*i

        i+=1

    return
# Multiple no.
n = 5;
# Number of whose multiple we are finding
k = 4;

print("Position of n\th multiple of k in"
      "Fibonacci Series is", findPosition(k,n));

```

```

# Q 4
c = 'p'
print("The ASCII value of '" + c + "' is", ord(c))

```

```

# Q 5
def squaresum(n):
    sm = 0
    for i in range(1, n+1):
        sm = sm + (i * i)

    return sm

```

```

n = 4
print(squaresum(n))

```

```

# Q 6
e = 8
f = 9
e = (e & f) + (e | f)
f = e + (~f) + 1
e = e + (~f) + 1
print("value of e after swapping : ",e)
print("value of f after swapping : ",f)

```

```

# Q 7
ch = input("Please Enter Your Own Character : ")

if((ch >= 'a' and ch <= 'z') or (ch >= 'A' and ch <= 'Z')):

```

```

    print("The Given Character ", ch, "is an Alphabet")
else:
    print("The Given Character ", ch, "is Not an Alphabet")

```

Q 8

```

def vowelOrConsonant(x):

    if (x == 'a' or x == 'e' or
        x == 'i' or x == 'o' or x == 'u'):
        print("Vowel")
    else:
        print("Consonant")

```

```

vowelOrConsonant('c')
vowelOrConsonant('e')

```

Q 9

```

ch = input("Please Enter Your Own Character : ")

if((ch >= 'a' and ch <= 'z') or (ch >= 'A' and ch <= 'Z')):
    print("The Given Character ", ch, "is an Alphabet")
elif(ch >= '0' and ch <= '9'):
    print("The Given Character ", ch, "is a Digit")
else:
    print("The Given Character ", ch, "is a Special Character")

```

Q 10

```

sub1=int(input("Enter marks of the first subject: "))
sub2=int(input("Enter marks of the second subject: "))
sub3=int(input("Enter marks of the third subject: "))
sub4=int(input("Enter marks of the fourth subject: "))
sub5=int(input("Enter marks of the fifth subject: "))
avg=(sub1+sub2+sub3+sub4+sub5)/5
if(avg>=90):
    print("Grade: A")
elif(avg>=80):
    print("Grade: B")
elif(avg>=70):
    print("Grade: C")
elif(avg>=60):
    print("Grade: D")
elif(avg>=40):
    print("Grade: E")
else:

```

```
print("Grade: F")
```

```
# Q. 11
```

```
basic_salary = float(input("Enter basic salary: "))
```

```
if basic_salary <= 10000:
```

```
    hra = 0.2 * basic_salary
```

```
    da = 0.8 * basic_salary
```

```
elif basic_salary <= 20000:
```

```
    hra = 0.25 * basic_salary
```

```
    da = 0.9 * basic_salary
```

```
else:
```

```
    hra = 0.3 * basic_salary
```

```
    da = 0.95 * basic_salary
```

```
gross_salary = basic_salary + hra + da
```

```
print(f"Gross salary is: {gross_salary}")
```

```
# Q. 12
```

```
# Input the electricity unit charges
```

```
unit_charges = int(input("Enter the electricity unit charges: "))
```

```
# Calculate the total electricity bill
```

```
if unit_charges <= 50:
```

```
    total_bill = unit_charges * 0.50
```

```
elif unit_charges <= 150:
```

```
    total_bill = 25 + ((unit_charges - 50) * 0.75)
```

```
elif unit_charges <= 250:
```

```
    total_bill = 100 + ((unit_charges - 150) * 1.20)
```

```
else:
```

```
    total_bill = 220 + ((unit_charges - 250) * 1.50)
```

```
# Add a 20% surcharge to the total bill
```

```
total_bill += (total_bill * 0.20)
```

```
# Print the total electricity bill
```

```
print(f"Total electricity bill is: {total_bill}")
```

```
# Q 13
```

```
def printLowercase():
```

```
    i = 0
```

```
    while i < 26:
```

```
        print(chr(97 + i), end = " ")
```

```
i = i + 1
```

```
printLowercase()
```

```
# Q 14
```

```
number = 1247
```

```
number = str(number)
```

```
first_digit = int(number[0])
```

```
last_digit = int(number[-1])
```

```
addition = first_digit + last_digit
```

```
print('Addition of first and last digit of the number is',  
      addition)
```

```
# Q 15
```

```
def getSum(n):
```

```
    sum = 0
```

```
    for digit in str(n):
```

```
        sum += int(digit)
```

```
    return sum
```

```
n = 12345
```

```
print(getSum(n))
```

```
# Q 16
```

```
num = int(input("enter a number"))
```

```
n = num
```

```
product = 1
```

```
while n != 0:
```

```
    rem = n % 10
```

```
    product = product * rem
```

```
    n = n // 10
```

```
print(product)
```

```
# Q 17
given_num = 12345

reverse_number = 0

while (given_num > 0):
    remainder = given_num % 10
    reverse_number = (reverse_number * 10) + remainder
    given_num = given_num // 10

print("The reversed number =", reverse_number)
```

```
# Q 18
import math

def rev(num):
    return int(num != 0) and ((num % 10) * \
        (10**int(math.log(num, 10))) + \
        rev(num // 10))
```

```
test_number = 9669669

print ("The original number is : " + str(test_number))

res = test_number == rev(test_number)

print ("Is the number palindrome ? : " + str(res))
```

```
# Q 19
num = int(input("Enter a number: "))

print("The factors of {} are,".format(num))

for i in range(1,num+1):
    if num % i == 0:
        print(i)
```

```
# Q 20
num = 7
factorial = 1

if num < 0:
    print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
```

```
    print("The factorial of 0 is 1")
else:
    for i in range(1,num + 1):
        factorial = factorial*i
    print("The factorial of",num,"is",factorial)
```

Q 21

```
import math
```

```
print("The gcd of 60 and 48 is : ", end="")
print(math.gcd(60, 48))
```

Q 22

```
def compute_lcm(x, y):

    if x > y:
        greater = x
    else:
        greater = y

    while(True):
        if((greater % x == 0) and (greater % y == 0)):
            lcm = greater
            break
        greater += 1

    return lcm
```

```
num1 = 54
```

```
num2 = 24
```

```
print("The L.C.M. is", compute_lcm(num1, num2))
```

Q 23

```
num = 11
```

```
if num > 1:
    for i in range(2, int(num/2)+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")
```

Q 24

```
upto = int(input("Find prime numbers upto : "))
```

```
print("\nAll prime numbers upto", upto, "are : ")
```

```
for num in range(2, upto + 1):
```

```
    i = 2
```

```
    for i in range(2, num):
```

```
        if(num % i == 0):
```

```
            i = num
```

```
            break;
```

```
    if(i != num):
```

```
        print(num, end=" ")
```

Q. 25

```
n = int(input("Enter a number: "))
```

```
sum = 0
```

```
for num in range(2, n+1):
```

```
    if all(num%i != 0 for i in range(2, int(num**0.5)+1)):
```

```
        sum += num
```

```
print("The sum of all prime numbers between 1 and", n, "is:", sum)
```

Q. 26

```
num = int(input("Enter a number: "))
```

```
# function to find all prime factors
```

```
def prime_factors(n):
```

```
    factors = []
```

```
    i = 2
```

```
    while i <= n:
```

```
        if n % i == 0:
```

```
            factors.append(i)
```

```
            n = n / i
```

```
        else:
```

```
            i += 1
```

```
    return factors
```

```
# printing the prime factors
```



```
print("Prime factors of", num, "are:", prime_factors(num))
```

```
# Q. 27
```

```
# take input from user
```

```
num = int(input("Enter a number: "))
```

```
# initialize sum and order variable
```

```
sum = 0
```

```
order = len(str(num))
```

```
# calculate sum of nth power of each digit
```

```
temp = num
```

```
while temp > 0:
```

```
    digit = temp % 10
```

```
    sum += digit ** order
```

```
    temp //= 10
```

```
# check if the number is Armstrong or not
```

```
if num == sum:
```

```
    print(num, "is an Armstrong number")
```

```
else:
```

```
    print(num, "is not an Armstrong number")
```

```
# Q. 28
```

```
n = int(input("Enter a number: "))
```

```
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)
```

```
# Q. 29
```

```
num = int(input("Enter a number: "))
```

```
divisors = []
```

```
for i in range(1, num):
```

```
    if num % i == 0:
```

```
        divisors.append(i)
```

```
if sum(divisors) == num:
    print(num, "is a perfect number")
else:
    print(num, "is not a perfect number")
```

Q. 30

```
num = int(input("Enter a number: "))
```

```
factorials = [1 if x == 0 else x * factorials[x - 1] for x in map(int, str(num))]
```

```
sum_factorials = sum(factorials)
```

```
if sum_factorials == num:
    print(f"{num} is a strong number")
else:
    print(f"{num} is not a strong number")
```

Q. 31

```
string = "racecar"
```

```
if string == string[::-1]:
    print("Palindrome")
else:
    print("Not Palindrome")
```

Q. 32

```
string = "Hello world"
```

```
words = string.split()
```

```
words = words[::-1]
```

```
new_string = " ".join(words)
```

```
print(new_string)
```

Q. 33

```
s = "example"
```

```
s = s[:2] + s[3:]
print(s)
```

```
s = s[1:]  
print(s)
```

```
s = s[:-1]  
print(s)
```

```
# Q. 34  
string = "hello world"  
substring = "world"
```

```
if substring in string:  
    print("Substring found in the string.")  
else:  
    print("Substring not found in the string.")
```

```
# Q. 35  
s = "this is a sample string with several words and some of these words are repeated"
```

```
word_freq = {word: s.count(word) for word in set(s.split())}  
  
print(word_freq)
```

```
# Q. 36  
snake_case_str = "my_foo_bar_variable"  
  
words = snake_case_str.split('_')  
pascal_case_str = ".join([word.capitalize() for word in words])  
  
print(pascal_case_str)
```

```
# Q. 37  
#Using the len() function:  
s = "Hello, World!"  
print(len(s))
```

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

```
s = "Hello, World!"  
print(sum(1 for char in s))
```

```
from functools import reduce
s = "Hello, World!"
print(reduce(lambda x, _: x+1, s, 0))
```

Q. 38

```
s = "This is a sample string containing several words with different lengths"
```

```
words = s.split()
```

```
for word in words:
    if len(word) % 2 == 0:
        print(word)
```

Q. 39

```
s = input("Enter a string: ")
vowels = {'a', 'e', 'i', 'o', 'u'}
```

```
if all(v in s.lower() for v in vowels):
    print(f"The string '{s}' contains all vowels.")
else:
    print(f"The string '{s}' does not contain all vowels.")
```

Q. 40

```
s1 = "Hello"
s2 = "World"
count = 0
```

```
for c in set(s1):
    count += min(s1.count(c), s2.count(c))
```

```
print(f"The number of matching characters in '{s1}' and '{s2}' is {count}.")
```

Q. 41

```
s = "Hello, World!"
s_unique = ".join(set(s))
print(s_unique)
```

Q. 42

```
from collections import Counter
```

```
string= "pppppgghhhijeupffe"
print(string)
```

```
result= Counter(string)
result= min(result, key=result.get)

print("Least frequent character: ",result)
```

Q. 43

```
s = "Hello, World!"
```

```
freq = {}
for c in s:
    freq[c] = s.count(c)
```

```
max_freq = max(freq.values())
max_char = [k for k, v in freq.items() if v == max_freq]
```

```
if len(max_char) == 1:
    print(f"The maximum frequency character in the string '{s}' is '{max_char[0]}', which appears {max_freq} times.")
else:
    print(f"There are multiple maximum frequency characters in the string '{s}', which are {', '.join(max_char)} and appear {max_freq} times each.")
```

Q. 44

```
import string
```

```
def has_special_chars(s):
    special_chars = set(string.punctuation)
    return any(char in special_chars for char in s)
```

```
s1 = "Hello, World!"
s2 = "Hello@World!"
```

```
print(has_special_chars(s1))
print(has_special_chars(s2))
```

Q. 45

```
s = "Hello, World!"
```

```
words = s.split()
```

```
joined_s = " ".join(words)
```

```
print(joined_s)
```

Q. 46

s1 = "hello world"

s2 = "world is beautiful"

s1_words = s1.split()

s2_words = s2.split()

s1_set = set(s1_words)

s2_set = set(s2_words)

uncommon_words = s1_set.symmetric_difference(s2_set)

print(uncommon_words)

Q. 47

s = "hello world"

new_s = ""

for i in range(len(s)):

if s[i] not in new_s:

new_s += s[i]

else:

new_s += ""

print(new_s)

#Q. 48

s = "hello world"

n = 3

rotated_s = s[n:] + s[:n]

print(rotated_s)

Q. 49

string = "hello world"

duplicates = {}

for char in string:

if char in duplicates:

duplicates[char] += 1

else:

duplicates[char] = 1

#Q 50

given_num = 12345

reverse_number = 0

while (given_num > 0):

 remainder = given_num % 10

 reverse_number = (reverse_number * 10) + remainder

 given_num = given_num // 10

print("The reversed number =", reverse_number)

import math

def rev(num):

 return int(num != 0) and ((num % 10) * \

 (10**int(math.log(num, 10)))) + \

 rev(num // 10))

#Q 51

test_number = 9669669

print ("The original number is : " + str(test_number))

res = test_number == rev(test_number)

print ("Is the number palindrome ? : " + str(res))

num = int(input("Enter a number: "))

print("The factors of {} are,".format(num))

for i in range(1,num+1):

 if num % i == 0:

 print(i)

#Q 52

num = /

factorial = 1

if num < 0:

 print("Sorry, factorial does not exist for negative numbers")

elif num == 0:

#Q 53

```
given_num = 12345
```

```
reverse_number = 0
```

```
while (given_num > 0):  
    remainder = given_num % 10  
    reverse_number = (reverse_number * 10) + remainder  
    given_num = given_num // 10
```

```
print("The reversed number =", reverse_number)
```

```
import math
```

```
def rev(num):  
    return int(num != 0) and ((num % 10) * \  
        (10**int(math.log(num, 10)))) + \  
        rev(num // 10)
```

#Q 54

```
test_number = 9669669
```

```
print ("The original number is : " + str(test_number))
```

```
res = test_number == rev(test_number)
```

```
print ("Is the number palindrome ? : " + str(res))
```

```
num = int(input("Enter a number: "))
```

```
print("The factors of {} are,".format(num))
```

```
for i in range(1,num+1):  
    if num % i == 0:  
        print(i)
```

#Q 55

```
num = /  
factorial = 1
```

```
if num < 0:  
    print("Sorry, factorial does not exist for negative numbers")  
elif num == 0:
```


#Q 56

```
upto = int(input("Find prime numbers upto : "))
```

```
print("\nAll prime numbers upto", upto, "are : ")
```

```
for num in range(2, upto + 1):
```

```
    i = 2
```

```
    for i in range(2, num):
```

```
        if(num % i == 0):
```

```
            i = num
```

```
            break;
```

```
    if(i != num):
```

```
        print(num, end=" ")
```

#Q 57

```
n = int(input("Enter a number: "))
```

```
sum = 0
```

```
for num in range(2, n+1):
```

```
    if all(num%i != 0 for i in range(2, int(num**0.5)+1)):
```

```
        sum += num
```

```
print("The sum of all prime numbers between 1 and", n, "is:", sum)
```

#Q 58

```
num = int(input("Enter a number: "))
```

```
# function to find all prime factors
```

```
def prime_factors(n):
```

```
    factors = []
```

```
    i = 2
```

```
    while i <= n:
```

```
        if n % i == 0:
```

```
            factors.append(i)
```

```
            n = n / i
```

```
        else:
```

```
            i += 1
```

```
    return factors
```

```
# printing the prime factors
```