Q1. Write a Python program that will print your name 10 times

In [1]:
```
print('Q1.\nRollno: 21052410\n')

for i in range(10):
    print('Brijit Adak')
```

```
Q1.
Rollno: 21052410

Brijit Adak
Brijit Adak
Brijit Adak
Brijit Adak
Brijit Adak
Brijit Adak
Brijit Adak
Brijit Adak
Brijit Adak
Brijit Adak
```

Q2. Write a Python program that will print 1 2 3 4 5 6 7 8 9 10

In [2]:
```
print('Q2.\nRollno: 21052410\n')

for i in range(10):
    print(i+1)
```

```
Q2.
Rollno: 21052410

1
2
3
4
5
6
7
8
9
10
```

Q3. Write a Python program that will print the number between m and n

In [3]:
```
print('Q3.\nRollno: 21052410\n')
import random
m=eval(input('Enter m: '))
n=eval(input('Enter n: '))

print(random.randint(m,n))
```

```
Q3.
Rollno: 21052410

Enter m: 4
Enter n: 7
6
```

Q4. Write a Python program that will print all odd number between m and n

In [7]:
```
print('Q4.\nRollno: 21052410\n')
m=eval(input('Enter m: '))
n=eval(input('Enter n: '))
for i in range(m,n):
    if(i%2!=0):
        print(i)
```

```
Q4.
Rollno: 21052410

Enter m: 3
Enter n: 9
3
5
7
```

Q5. Write a Python program that will print 9 7 5 3 1 -1 -3 -5 -7 -9

In [8]:
```
print('Q5.\nRollno: 21052410\n')
for i in range(9,-10,-2):
    print(i,end=" ")
```

```
Q5.
Rollno: 21052410

9 7 5 3 1 -1 -3 -5 -7 -9
```

Q6. Write a Python program that will print sum of the following series Sum = 1+ ½ + 1/3 + …..1/n

In [10]:
```
print('Q6.\nRollno: 21052410\n')
n=eval(input("Enter n: "))
s=0;
for i in range(n):
    s+=1/(i+1)
print('Sum = ',s)
```

```
Q6.
Rollno: 21052410

Enter n: 2
Sum =  1.5
```

Q7 Write a Python program that will print sum of the following series Sum = 1+ ½! + 1/3! + …..1/n!

In [11]:
```
print('Q7.\nRollno: 21052410\n')
n=eval(input("Enter n: "))
s=0;
for i in range(n):
    f=1
    for j in range(i+1):
        f*=(j+1)
    s+=1/f
print('Sum = ',s)
```

```
Q7.
Rollno: 21052410

Enter n: 2
Sum =  1.5
```

```
Q8. Write a Python program that will print sum of the following series
e^x = 1+ x + x^2/2! +x^3/3! …..x^n/n!
```

In [13]:
```
print('Q8.\nRollno: 21052410\n')
import math

def calculate_series_sum(x, n):
    series_sum = 0

    for i in range(n + 1):
        term = (x ** i) / math.factorial(i)
        series_sum += term

    return series_sum

x_value = float(input("Enter the value of x: "))
n_terms = int(input("Enter the number of terms (n): "))

result = calculate_series_sum(x_value, n_terms)
print(f"The sum of the series e^{x_value} is: {result}")
```

```
Q8.
Rollno: 21052410

Enter the value of x: 3
Enter the number of terms (n): 7
The sum of the series e^3.0 is: 19.846428571428568
```

Q9. Write a Python program that will read x and compute sin(x) (Hints: Use Taylor's series expansion)

In [14]:
```
print('Q9.\nRollno: 21052410\n')
import math

def compute_sin(x, num_terms=10):
    result = 0
    for n in range(num_terms):
        term = ((-1) ** n) * (x ** (2 * n + 1)) / math.factorial(2 * n
        result += term
    return result

degrees = float(input("Enter the angle in degrees: "))
x_radians = math.radians(degrees)

num_terms = 10

sin_x = compute_sin(x_radians, num_terms)

print(f"The sine of {degrees} degrees is approximately: {sin_x}")
```

```
Q9.
Rollno: 21052410

Enter the angle in degrees: 45
The sine of 45.0 degrees is approximately: 0.7071067811865475
```

Q10. Write a Python program that will read x and compute cos(x) (Hints: Use Taylor's series expansion)

In [15]:
```
print('Q10.\nRollno: 21052410\n')
import math

def compute_cos(x, num_terms=10):
    result = 0
    for n in range(num_terms):
        term = ((-1) ** n) * (x ** (2 * n)) / math.factorial(2 * n)
        result += term
    return result

degrees = float(input("Enter the angle in degrees: "))
x_radians = math.radians(degrees)
num_terms = 10

cos_x = compute_cos(x_radians, num_terms)

print(f"The cosine of {degrees} degrees is approximately: {cos_x}")
```

```
Q10.
Rollno: 21052410

Enter the angle in degrees: 45
The cosine of 45.0 degrees is approximately: 0.7071067811865475
```

Q11 Write a Python program that will check the number is prime or composite.

In [16]:
```
print('Q11.\nRollno: 21052410\n')
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

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

if is_prime(num):
    print(f"{num} is a prime number.")
else:
    print(f"{num} is a composite number.")
```

```
Q11.
Rollno: 21052410

Enter a number: 23
23 is a prime number.
```

Q12. Write a Python program that will read two integers and compute GCD and LCM.

In [17]:

```python
print('Q12.\nRollno: 21052410\n')
def compute_gcd(x, y):
    while y:
        x, y = y, x % y
    return abs(x)

def compute_lcm(x, y):
    return abs(x * y) // compute_gcd(x, y)

num1 = int(input("Enter the first integer: "))
num2 = int(input("Enter the second integer: "))

gcd_result = compute_gcd(num1, num2)
lcm_result = compute_lcm(num1, num2)

print(f"The GCD of {num1} and {num2} is: {gcd_result}")
print(f"The LCM of {num1} and {num2} is: {lcm_result}")
```

```
Q12.
Rollno: 21052410

Enter the first integer: 20
Enter the second integer: 5
The GCD of 20 and 5 is: 5
The LCM of 20 and 5 is: 20
```

Q13. Write a Python program that read an integer and print the number of digit.

In [18]:

```python
print('Q13.\nRollno: 21052410\n')
def count_digits(number):
    return len(str(number))

if __name__ == "__main__":

    try:
        user_input = int(input("Enter an integer: "))
        num_digits = count_digits(user_input)
        print(f"The number of digits in {user_input} is: {num_digits}")

    except ValueError:
        print("Invalid input. Please enter a valid integer.")
```

```
Q13.
Rollno: 21052410

Enter an integer: 124
The number of digits in 124 is: 3
```

Q14. Write a Python program that will read a number and compute sum of the digit Ex: let num= 3456 output should be 18

In [19]:
```python
print('Q14.\nRollno: 21052410\n')
def sum_of_digits(number):
    return sum(int(digit) for digit in str(number))

if __name__ == "__main__":
    try:
        user_input = int(input("Enter an integer: "))
        digit_sum = sum_of_digits(user_input)
        print(f"The sum of digits in {user_input} is: {digit_sum}")

    except ValueError:
        print("Invalid input. Please enter a valid integer.")
```

```
Q14.
Rollno: 21052410

Enter an integer: 3456
The sum of digits in 3456 is: 18
```

Q15. Write a Python program that will reverse an integer i.e num =3456 reverse num=6543

In [20]:
```python
print('Q15.\nRollno: 21052410\n')

def reverse_integer(number):
    reversed_number = int(str(number)[::-1])
    return reversed_number
user_input = input("Enter an integer: ")

try:
    user_number = int(user_input)
    reversed_result = reverse_integer(user_number)

    print(f"The reversed integer is: {reversed_result}")

except ValueError:
    print("Invalid input. Please enter a valid integer.")
```

```
Q15.
Rollno: 21052410

Enter an integer: 3456
The reversed integer is: 6543
```

Q16. Write a Python program that will check a number is palindrome or not. i.e 12321 is a palindrome

In [21]:
```
print('Q16.\nRollno: 21052410\n')
def is_palindrome(number):
    str_number = str(number)
    return str_number == str_number[::-1]

user_input = input("Enter a number: ")

try:
    user_number = int(user_input)
    if is_palindrome(user_number):
        print(f"{user_number} is a palindrome.")
    else:
        print(f"{user_number} is not a palindrome.")

except ValueError:
    print("Invalid input. Please enter a valid number.")
```

```
Q16.
Rollno: 21052410

Enter a number: 12321
12321 is a palindrome.
```

Q17. Write a python program to find the Fibonacci series up to nth term.

In [22]:
```
print('Q17.\nRollno: 21052410\n')
def fibonacci_series(n):
    fib_series = [0, 1]

    while len(fib_series) < n:
        next_term = fib_series[-1] + fib_series[-2]
        fib_series.append(next_term)

    return fib_series[:n]

try:
    n = int(input("Enter the number of terms for the Fibonacci series:
    if n <= 0:
        raise ValueError("Please enter a positive integer.")
    result = fibonacci_series(n)
    print(f"Fibonacci series up to the {n}th term: {result}")

except ValueError as e:
    print(f"Error: {e}")
```

```
Q17.
Rollno: 21052410

Enter the number of terms for the Fibonacci series: 5
Fibonacci series up to the 5th term: [0, 1, 1, 2, 3]
```

Q18. Write a Python program to check the number is Armstrong or not. For example, 371 is an Armstrong number since 333 + 777 + 111 = 371.

In [24]:

```
print('Q18.\nRollno: 21052410\n')
def is_armstrong_number(number):
    num_str = str(number)
    num_digits = len(num_str)
    armstrong_sum = sum(int(digit) ** num_digits for digit in num_str)
    return armstrong_sum == number

user_input = input("Enter a number: ")

try:
    user_number = int(user_input)

    if is_armstrong_number(user_number):
        print(f"{user_number} is an Armstrong number.")
    else:
        print(f"{user_number} is not an Armstrong number.")

except ValueError:
    print("Invalid input. Please enter a valid number.")
```

```
Q18.
Rollno: 21052410

Enter a number: 371
371 is an Armstrong number.
```

Q19. Write a Python program that will display the prime’s number between M and N.

In [25]:

```python
print('Q19.\nRollno: 21052410\n')
def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

def display_primes_between_m_and_n(m, n):
    primes = [num for num in range(m, n+1) if is_prime(num)]
    return primes

try:
    m = int(input("Enter the value of M: "))
    n = int(input("Enter the value of N: "))

    if m < 0 or n < 0:
        raise ValueError("Please enter non-negative integers.")

    if m > n:
        raise ValueError("M should be less than or equal to N.")

    result = display_primes_between_m_and_n(m, n)
    print(f"Prime numbers between {m} and {n}: {result}")

except ValueError as e:
    print(f"Error: {e}")
```

```
Q19.
Rollno: 21052410

Enter the value of M: 3
Enter the value of N: 9
Prime numbers between 3 and 9: [3, 5, 7]
```

Q20. Write a python program that will take a positive integer (num say) as input and display the positive numbers, which are less than num and relatively prime to num.

In [26]:
```
print('Q20.\nRollno: 21052410\n')
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a

def relatively_prime_numbers(num):
    return [i for i in range(1, num) if gcd(num, i) == 1]

try:
    num = int(input("Enter a positive integer (num): "))

    if num <= 0:
        raise ValueError("Please enter a positive integer.")

    result = relatively_prime_numbers(num)
    print(f"Positive numbers less than {num} that are relatively prime

except ValueError as e:
    print(f"Error: {e}")
```

```
Q20.
Rollno: 21052410

Enter a positive integer (num): 12
Positive numbers less than 12 that are relatively prime to 12: [1, 5,
7, 11]
```

Q21. Write python programs that will print the following output.
a.
```
1
1 2
1 2 3
1 2 3 4
```

In [27]:
```
print('Q21->a.\nRollno: 21052410\n')
def print_pattern_a(rows):
    for i in range(1, rows + 1):
        for j in range(1, i + 1):
            print(j, end=" ")
        print()

print_pattern_a(4)
```

```
Q21->a.
Rollno: 21052410

1
1 2
1 2 3
1 2 3 4
```

b.
```
1
2 2
3 3 3
```

```
4 4 4 4
```

In [28]:

```python
print('Q21->b.\nRollno: 21052410\n')
def print_pattern_b(rows):
    for i in range(1, rows + 1):
        for j in range(1, i + 1):
            print(i, end=" ")
        print()

print_pattern_b(4)
```

```
Q21->b.
Rollno: 21052410

1
2 2
3 3 3
4 4 4 4
```

```
c.
A A A A A A
A         A
A         A
A         A
A A A A A A
```

In [30]:

```python
print('Q21->c.\nRollno: 21052410\n')
def print_pattern_c(rows, columns):
    for i in range(rows):
        for j in range(columns):
            if i == 0 or i == rows - 1 or j == 0 or j == columns - 1:
                print("A", end=" ")
            else:
                print(" ", end=" ")
        print()

print_pattern_c(5, 6)
```

```
Q21->c.
Rollno: 21052410

A A A A A A
A         A
A         A
A         A
A A A A A A
```

```
d.
a
a b
a b c
```

In [31]:

```
print('Q21->d.\nRollno: 21052410\n')
def print_pattern_d(rows):
    for i in range(rows):
        for j in range(i + 1):
            print(chr(ord('a') + j), end=" ")
        print()

print_pattern_d(3)
```

```
Q21->d.
Rollno: 21052410

a
a b
a b c
```

```
e.
 aaaaaa
 aaaaa
 aaaa
 aaa
 aa
 a
```

In [32]:

```
print('Q21->e.\nRollno: 21052410\n')
def print_pattern_e(rows):
    for i in range(rows):
        print(" " * i + "a" * (rows - i))

print_pattern_e(6)
```

```
Q21->e.
Rollno: 21052410

aaaaaa
 aaaaa
  aaaa
   aaa
    aa
     a
```

```
f.
 *
 * * *
 * * * * *
 * * * * * * *
 * * * * * * * * *
```

In [33]:
```
print('Q21->f.\nRollno: 21052410\n')
def print_pattern_f(rows):
    for i in range(rows):
        print(" " * (2 * (rows - i - 1)) + "* " * (2 * i + 1))

print_pattern_f(5)
```

```
Q21->f.
Rollno: 21052410

        *
      * * *
    * * * * *
  * * * * * * *
* * * * * * * * *
```

```
g.
        1
      2 3 2
    3 4 5 4 3
  4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

In [34]:
```
print('Q21->g.\nRollno: 21052410\n')
def print_pattern_g(rows):
    for i in range(1, rows + 1):
        print(" " * (2 * (rows - i)), end="")

        for j in range(i, 2 * i):
            print(j, end=" ")

        for j in range(2 * i - 2, i - 1, -1):
            print(j, end=" ")

        print()

print_pattern_g(5)
```

```
Q21->g.
Rollno: 21052410

        1
      2 3 2
    3 4 5 4 3
  4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
```

```
h.
      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
```

```
1 5 10 10 5 1
```

In [35]:

```python
print('Q21->h.\nRollno: 21052410\n')
def print_pattern_h(rows):
    for i in range(rows):
        print(" " * (2 * (rows - i)), end="")
        for j in range(0, i + 1):
            print(binomial_coefficient(i, j), end="   ")

        print()

def binomial_coefficient(n, k):
    result = 1
    if k > n - k:
        k = n - k
    for i in range(k):
        result *= (n - i)
        result //= (i + 1)
    return result

print_pattern_h(6)
```

```
Q21->h.
Rollno: 21052410

            1   
          1   1   
        1   2   1   
      1   3   3   1   
    1   4   6   4   1   
  1   5   10   10   5   1   
```