Name : Abdullah Asif

Class : BSCS 2nd A.

Roll no : 2023108

Submitted to : Jamal Abdul ahad



1. Write a countdown iterator.

Program:

class CountdownIterator: def \_init\_(self, start):

self.current = start

def \_iter\_(self): return self

def \_next\_(self):

if self.current <= 0: raise StopIteration else:

self.current -= 1 return self.current + 1



2. Create an iterator to iterate over a string in python.

Program:

class StringIterator: def \_init\_(self, input\_string):

self.input\_string = input\_string self.index = 0

def \_iter\_(self): return self

def \_next\_(self): if self.index < len(self.input\_string): result = self.input\_string[self.index] self.index += 1 return result else:

raise StopIteration

# Example usage:

string\_iterator = StringIterator("Hello, World!")

for char in string\_iterator:

print(char)



3. Create an iterator that iterates over power of 2 for a given range.

Program:

def powers\_of\_two\_iterator(start, end): current\_power = 0

current\_value = 2 \*\* current\_power

while current\_value <= end: if current\_value >= start: yield current\_value current\_power += 1

current\_value = 2 \*\* current\_power

# Example usage: start\_value = 1 end\_value = 64

for power\_of\_two in powers\_of\_two\_iterator(start\_value, end\_value):

print(power\_of\_two)



4. Implement a custom iterator that calculates prime numbers upto a given range.

Program:

class PrimeNumbersIterator:

def \_init\_(self, end): self.end = end self.current = 2

def \_iter\_(self): return self

def is\_prime(self, num):

for i in range(2, int(num\*\*0.5) + 1): if num % i == 0: return False return True

def \_next\_(self): while self.current <= self.end: if self.is\_prime(self.current):

result = self.current self.current += 1 return result else:

self.current += 1

raise StopIteration

# Example usage:

prime\_iterator = PrimeNumbersIterator(30)

for prime\_number in prime\_iterator: print(prime\_number)

