**Data structures and Alogrithm Lab**

****

|  |  |
| --- | --- |
| Name | **Talha Khalil** |
| Registration no. | **200901003** |
| Batch & Section | **2020 BS-Computer Science section A** |
| Instructor’s name | **Sir Nadeem** |

**Lab task: 01**

**Write a program to implement queue using numpy array.**

**Code:-**

# Implementation of queue using numpy array

import numpy as np

class queue:

    def \_init\_(self, size):

        self.size = size

        self.elements = np.zeros([self.size], dtype='int8')

        self.clear()

    def enqueue(self, data):

        if self.isFull():

            print("---Queue Overflow---")

        else:

            self.rear = self.rear + 1

            if self.front == -1:

                self.front = 0;

            self.elements[self.rear] = data

    def dequeue(self):

        if self.isEmpty():

            return "---Queue Underflow---"

        else:

            value = self.elements[self.front]

            self.front = self.front + 1

            return value

    def isEmpty(self):

        return ((self.front == -1 or self.rear == -1) or (self.front > self.rear))

    def isFull(self):

        return (self.rear == self.size - 1)

    def clear(self):

        self.rear = -1

        self.front = -1

q = queue(7)

q.enqueue(9)

q.enqueue(2)

print(q.dequeue())

print(q.dequeue())

print(q.dequeue())

**Lab task: 02**

**Write documentation of deque from Collections.**

append(x)¶ Add x to the right side of the deque.

appendleft(x) Add x to the left side of the deque.

clear() Remove all elements from the deque leaving it with length 0.

copy() Create a shallow copy of the deque.

count(x) Count the number of deque elements equal to x.

extend(iterable) Extend the right side of the deque by appending elements from the iterable argument.

extendleft(iterable) Extend the left side of the deque by appending elements from iterable. Note, the series of left appends results in reversing the order of elements in the iterable argument.

index(x[, start[, stop]]) Return the position of x in the deque (at or after index start and before index stop). Returns the first match or raises ValueError if not found.

insert(i, x) Insert x into the deque at position i.

If the insertion would cause a bounded deque to grow beyond maxlen, an IndexError is raised.

pop() Remove and return an element from the right side of the deque. If no elements are present, raises an IndexError.

popleft() Remove and return an element from the left side of the deque. If no elements are present, raises an IndexError.

remove(value) Remove the first occurrence of value. If not found, raises a ValueError.

reverse() Reverse the elements of the deque in-place and then return None.

rotate(n=1) Rotate the deque n steps to the right. If n is negative, rotate to the left.

When the deque is not empty, rotating one step to the right is equivalent to d.appendleft(d.pop()), and rotating one step to the left is equivalent to d.append(d.popleft()).

Deque objects also provide one read-only attribute:

maxlen Maximum size of a deque or None if unbounded.