

LABORATORY WORK BOOK

Name of the Student Muhd Fauzan Zahaih						Roll Number			
Class D tech Cot C Semester A						2 3 9	5 1 A C	5	5 #
Nino	on of the	Course Faculty Da M M	adhusud	han M	ddy		Faculty ID : !!	ARE I	0881
Exe	rcise Nu	mber	Week	Number :					
S. No.	Exercise Number	EXERCISE NAME	Aim/ Preparation Performance in the Lab		Source Code Calculations and Graphs	Program Execution Results and Error Analysis	Viva - Voce	Total	
			4	4		4	4	4	20
1	9.1	Lineou Queux using	5		100		ere sesse	100	
2	5-2	stack using queues			Eart (E)		n JERGIE		
3	5.3	Implement queue	74	4		4	4	4	26
4		using stacks	oofi,	: Joseph	geori de	e wat t	y July		
5	5-4	ciscular queue.					# N A	7.60	
6				59300	000 - 000	oli oli	1	4	
7	Sec.	enditon.		73		S. Carlan	and the		
8					Samuel.	a land	i ha	R Lt	
9		Capped Phone					1- Front		
10		Tot lot lat	Mark	Laur	was t	07500 t (.)	1 hata		
11								. 11	Y
12			T.E.		1				

Signature of the Student

Signature of the Faculty

1

```
4.1 Linear Queue
-: woodsaw:
front = a
xean = a
my max = 5
det coeate Queue ():
      queue = [] # empty
     retwin queue
def is Empty (queue):
     seturn len (queue) = =0
det enqueue Cqueue, item):
   global rear, my max
    if reax my max;
        queue append (item)
        260H + = 1
        print (" Element enqueued: " sitem)
else :
    print (" queue is full. Element cannot be enqueued.")
det degueue (queue):
    global front, rear
 if not is Empty (queue)
       front += 1
       print (" Element de queued - queue, pop(0))
else:
  print ("queue is empty comnot dequeue; ")
# Driver code
queue = (oreate queue ()
while Toue:
                               2/16
```

```
print (" 2 Enqueue ")
  print ( "2. orqueue")
  point ("3. Display")
   print ("4- quit")
   choice = int (input (" Inter your chaice: "))
if choice = = 1 :
     item = input (" Enter elements to enqueue :")
     emajueuc (queue, item)
elif choice = = 2 :
     dequeue (queue)
elit choice = = 3;
     point ("queue; ", queue)
elif choice = = 4:
      print ("Exiting --- ")
      breat.
  Print ("Invalid choice. Please enter a valid aption ")
                              "phone his and puty books
Exercise 4.2. stack using queue >
5xodsaw:
                                ( when states ) tuget a soled
from collection impost dequeue
class stack using queue:
                      water - input Children when to proper - solar
    det_int_ (self):
        self - queue = dequeue ()
    def push (self, x: sto) -> None:
        self queue append (2)
    for i in nange (lem (self gueue)-1):
self. que ue append (self. queue popleft())
                              3/16
```

```
det pop (self) - sto:
  if self-queue:
     setwin self-queue poplett()
  else:
    print ("stack is empty, camnot pop")
     seturn None
 def top(self) -> sto:
   it self quelle
      seture self - queue [0]
      print ("stack is empty. No top element")
   else:
     return None
def Empty (self) -> bool:
    seturn len(self.queue) == 0
stack = stack using queue ()
while Toue:
      print ("1-push")
     print (112- pop11)
  print ("4. check if empty")
                       Could was short wing queue x
     print ("5. txit")
choice = input ("Entex choice")
                        Two to proper deported
if choice = = 'i';
                              TENTON TOTAL
    value = input ("Inter value to push: ")
    stack push (value)
elif choice = = 12';
    P=stack. POP()
if p is not none:
     print (" popped value : P)
                        4/16
```

```
elit choice on 3' I . Il topic young . Fin't donny que to spell, the
  t = stark -top() (3 qot - Araba = t
is not work if H
  to not want ; " to p element ; " t)
elit chaice = = '4's
  if stack empty ()
     point ("Tope stack is emply ")
     else:
  elit chake = = 15"
      point ("Emiting -- ") = 357 12 37 town
 else:
    point ("Invalid choice")
                in alor who "I depot to when roller to do
4.3 Implement Queue using stack:
program :-
class queue using stack:
    det -- Init --- Coelf): [ me of service ) thing
    self. stock = []
    self . stack 2 - []
det emqueue (sett, as int) -> None;
    self. stack 2. appendicx)
det dequeue (self) -> int:
   if not self. stack 2:
       while self. stack 1:
self . stack 2:
     while self . stack 1:
                      5/16
```

```
self stack 2 append (self . stack 1. pop())
    setwin. self. stack 2. pop()
def is - Empty (self) -> bool:
     return not self stacks and not self stocke
queue = queue using stack ()
while True:
     print (" 1. Enqueue ")
      Dajut (11 5. Dedriene 11)
    point (" 3. check it empty")
      print (" 4. Git")
choice = input ()
if choice = = "1"; value = int (input (" enter value to Engueue: "))
queue enqueue (value) de poiser sono de la manda de la
    elif choice = = 121:
        if queue is - empty ()"
                                idonto polar sum a contr
            print (" queue is empty: cannot dequeue ")
 · e/se:
      dequeue = queue dequeue ()
 point print ("queue is empty · cannot dequeue")
                              (c) draggo foots me
 else :
     d q: quare. dequeue ()
     point (" Dequeued values : " deq )
 elif choice = = 13':
                         6/16
```

```
it queue_15-empty ();
     print ("queue is empty")
else:
    point ("queue is not empty")
elif choice = = 141:
                                      mir him - found the fi
print (" Exiting .... ")
break
   point ("Involid choice") Was all the delibert that the
else:
                             La . " I taranta Lauragent " > hing
4.4 clocular queue:
                                                   Z charles
brodraw ?
class circular queue;
    def_init_ (self, capacity):
     self. capacity capacity
     self queue = [None] * capacity
     self. front = self. reax == 1
                                       O tuqui 4
det Enqueue (sett, x):
     it (self. sear + 1) 7. self. capacity = = self. fourts
  print ("queue is full! cannot Enqueue ")
   return
elit self. front == -1
     self foont =0
self. sear = (self. sean + Dy. self capacity
     self · que ue (self · rear) = x
     point ("Enqueue element: 7/16 1x).
```

```
det dequeue (self):
   if self. front = = -1
print ("quede is Empty")
   seturn wone Coppen to I some of the
x = self queue (self foont)
it self front = self · rear
                                (" pathirs") long
 self. front = self. Mean = -1
else :
 self-foont = (self-foont +1) % self-capacity
point (" Dequeued element: ", x)
                            4. 21 Median Curille &
setwin x
capacity = 'mt (input ())
queue = circular queue (capacity)
                           olytimper . Had - their tab
while True:
     point (" 1 Englueue ")
      point (" 2. Dequeue")
      print (" 3. Exit ")
        e = input ()
     it comput ()
     it e== " 5 " surport tomes 1 int
 e = input ("Enter element to enqueue; ")
  queue = enqueue (e)
elif c = = 2':
    que ne dequeue (°)
                         X x CMXX HOLD SUGAR + FIX
elit c=3';
                    Ala 8/16 Thomas Tong Carl I bring
```

point (" Exiting --- ") break. else : point ("Invalid choice"). (17 the Belleville of a traged self green to be a selfect ((Knommer) ms) spower of fresh (0) 404.0 (' smail bring LIDEA SERVE SHIPE , Print of 9/16

```
Linear queue
401
beodaau.
m = int (input ())
command - list (map (sto, input (). split (; )))
values = list (map (sto, input () . split (', 1)1)
OV = []
for i in slange (len (command));
     if command [i] = = |add |:
           ar. append (values[i])
     elif command [i] == size :
     print (len(q1))
     elif command [i] == 'dequeue';
       q. pop (0)
      elif command [i] == 'print':
        for j in ay;
             print (ij end = 1 1)
 Input
 6
add, add, add, size, degrueue, print
1, 3, 4, NULL, NULL, NULL
 output
accepted.
```

10/16

```
4-2 Stack using queues
quize = int (input ())
Size = []
command = list (map (sts, input (), split (', ')))
value = list (map (str., input () . split (', '))) formag
for 1 in Hange (len (command)): 11 15 person
     if command [i] = = 'add';
       if len(s) < queue:
                               sympt slinke
            5. append (value [i])
         else:
             print ('queue is full')
     clif command [[] = = 'pop';
         s.pop ()
     elif command [i] = = Isize!; fort
                                      (d) Higg
         point (len(s))
     elif command [i] = = 'print':
                                            I hogai
           point (s(lents),-1-j], end = '1)
         for i in nonge (len ();
input :
add, pop, add, size, point, pop
I, NULL, 3, NULL, NULL
output
accepted.
```

11/16

```
4.3 Implement queue using stack he miss doubt the
Dag gram +
n = int (input ())
it n == 0: (( + ) ) tings () togal at ) qualtail , bisconnic
point ("6") ") High () togat, stell grows toll - soles
for 1 in range (1, n+1) & months made of
    b = ""
    temp = 9
                     the least of the state of
    while temp: (1) solar tongo.
    it temp % 2 == 1:
         b = 12 ( thebel a MADD ) tollog
         else:

b = 101 + b 1000 = 111 bournous tile
       temp = temp 11. 2
    point (b)
                  : third == (1) promines the
 input &
          box (Ci-1-Co) (1) or to figet
                                       - chant
                     spop hours ser bbo gog bbo
 10
                            THURST F. JUNE . D.
                       12/16
```