**“ QUEUE IN PYTHON ”**

**LAB WORK:**

**1.Program**

class Queue:

def \_\_init\_\_ (self):

self.items=[]

def isEmpty(self):

return self.items==[]

print(self)

def enqueue(self,item):

self.items.insert(0,item)

print(item)

def dequeue(self):

return self.items.pop()

def size(self):

return len(self.items)

q=Queue()

q.enqueue(5)

q.enqueue('dog')

q.enqueue('true')

print(q.size())

print(q.items)

q.dequeue()

print(q.items)

**OUTPUT**

5

dog

true

3

['true', 'dog', 5]

['true', 'dog']

**2.Program**

import queue

l=queue.Queue(maxsize=20)

l.put(3)

l.put(9)

l.put(1)

l.put(7)

print(l.qsize())

for a in range(20):

print(l.get(a))

**OUTPUT**

4

3

9

1

7

**3.Program**

import queue

l=queue.LifoQueue()

for a in range(4):

l.put("item "+str(a))

while not l.empty():

print(l.get())

**OUTPUT**

item 3

item 2

item 1

item 0

**4.Program**

import queue as q

q=q.PriorityQueue()

for a in range(4):

q.put(str(a))

while not q.empty():

print(q.get())

**OUTPUT**

0

1

2

3

**5.Program**

import heapq

heap=[]

heapq.heappush(heap,(4,'G'))

heapq.heappush(heap,(2,'A'))

heapq.heappush(heap,(3,'B'))

heapq.heappush(heap,(1,'C'))

while heap:

item=heapq.heappop(heap)

print(item)

**OUTPUT**

(1, 'C')

(2, 'A')

(3, 'B')

(4, 'G')

**EXERCISE:**

**Object 01:**Implement LIFO queue without using build in queue class.

**Source Code:**

class Queue:

def \_\_init\_\_ (self):

self.items=[]

def isEmpty(self):

return self.items==[]

print(self)

def enqueue(self,item):

self.items.insert(2,item)

print(item)

def dequeue(self):

return self.items.pop()

def size(self):

return len(self.items)

q=Queue()

q.enqueue(5)

q.enqueue('dog')

q.enqueue('true')

print(q.size())

print(q.items)

q.dequeue()

print(q.items)

**Output:**

5

dog

true

3

[5, 'dog', 'true']

[5, 'dog']

**Object 02:**We have an array of 5 elements [4,8,1,7,3] and we have to insert all the elements in priority queue.The output should by :

4

8 4

8 4 1

8 7 4 1

8 7 4 3 1

**Source Code:**

import queue as q

a=[4,8,1,7,3]

q=q.PriorityQueue()

for l in range(6):

q.put((l,a[l]))

if(l==0):

print(a[l])

m=l+1

if(a[m]>a[l]):

print(str(a[m])+" "+str(a[l]))

n=m+1

if(a[m]>a[l]) & (a[m]>a[n]) & (a[l]>a[n]):

print(str(a[m])+" "+str(a[l])+" "+str(a[n]))

y=n+1

if(a[m]>a[l]) & (a[m]>a[n]) & (a[l]<a[y]) & (a[l]>a[n]):

print(str(a[m])+" "+str(a[y])+" "+str(a[l])+" "+str(a[n]))

z=y+1

if(a[m]>a[l]) & (a[m]>a[n]) & (a[l]<a[y]) & (a[l]>a[n]) & (a[z]<a[m]):

print(str(a[m])+" "+str(a[y])+" "+str(a[l])+" "+str(a[z])+" "+str(a[n]))

**Output:**

4

8 4

8 4 1

8 7 4 1

8 7 4 3 1

**Object 03:**

Initailize marks and student roll no in heap and print the students roll no according to marks from heighest to lowest.

**Source Code:**

import heapq

maxheap=[]

heapq.heappush(maxheap,(65,09))

heapq.heappush(maxheap,(73,11))

heapq.heappush(maxheap,(55,15))

heapq.heappush(maxheap,(40,07))

while (maxheap):

item=heapq.heappop(maxheap)

print(‘Roll no: ’,item[1])

**Output:**

Roll no: 11

Roll no: 09

Roll no: 15

Roll no: 07