Class Queue:

Def \_\_init\_\_(self,max\_size):

Self.max\_size = max\_size

Self.Q = [0] \* max\_size

Self.num = 0

Self.first = 0

Def enqueue (self,item):

If self.num >= self.max\_size:

Raise exception (“Queue overflow”)

Self.Q[(self.num + self.first)%self.max\_size]=item

Self.num += 1

Def dequeue (self):

If self.num == 0:

Raise exception(“Queue empty”)

Item = self.Q[self.first]

Self.first = (self.first + 1 ) % self.max\_size

Self.num -= 1

Return item

Def front(self):

If self.num == 0:

Raise exception (“Queu empty”)

Return self.Q[self.first]

Def is\_empty(self):

Return self.num == 0

Def size(self):

Return self.num

Def is\_full(self):

Return self.num >= self.max\_size

Def delete (self,x):

If x <= self.num:

Item1 = self.Q[x]

Item2 = x+1

Self.Q = self.Q[:x] + self.Q[item2: ]

Else:

Raise exception(“error”)

Return self.Q,item1