

```
class QueueUsingStack :  
    let s1 and s2 be two stacks  
    method enqueue(x):  
        while(s1 is not empty) :  
            s2.push(s1.pop())  
        s1.push(x)  
        while(s2 is not empty) :  
            s1.push(s2.pop())  
  
    method dequeue () :  
        if (s1.isEmpty() ):  
            print("queue is empty")  
            return null  
        return s1.pop()
```

```
class StackUsingQueue :  
    let q be a Queue  
    method push(x):  
        n = q.size()  
        q.enqueue(x)  
        repeat n times :  
            q.enqueue(q.dequeue())  
  
    method pop() :  
        if (q.isEmpty() ):  
            print("stack is empty")  
            return null  
        return q.dequeue()
```

## 1- recursive

method reverseQueue(q) :

if ( q.isEmpty() )

return

p = q.dequeue()

reverseQueue(q)

q.enqueue(p)

## 2- iterative using stack :

method reverseQueue(q) :

s = stack()

while ( !q.isEmpty() )

s.push(q.dequeue() )

while ( !s.isEmpty() )

q.enqueue( s.pop() )

let A[1...n] be new array

top1 = 0

top2 = n+1

function isFull()

if top2 = top1 + 1 :

return True

else :

return False

function push(x,s)

if isFull() :

print("stack is Full")

else if s==1:

A[++top1] = x

else :

A[--top2] = x

function pop(s)

x = -1

if s==1 :

if top1>0 :

x=A[top1--]

else :

print("stack1 is empty")

else if s==2 :

if top2<=n :

x=A[top2++]

else :

print("stack2 is empty")

return x

بخش امتیازی سوال پشتک :

این بخش را با سه تا آرایه میتوان حل کرد.

یک آرایه برای دیتای پشته‌ها. یکی برای اندیس top هر پشته و یکی هم برای اندیس next هر پشته.

موقع push در پشته نام کافیسست که  $top[i]$  ، اولین خانه خالی آرایه اول قرار داده شود و  $next[i]$  هم  $top[i]$  قبلی شود.

برای pop هم  $top[i]=next[i]$  و عنصری که در  $top[i]$  قبلی بود برگردانده می شود.