

# स्टैक और क्यू

---

## पाठ्यपुस्तक के प्रश्न

### वस्तुनिष्ठ प्रश्न

**प्रश्न 1.** निम्न में से कौन-सा नाम स्टैक से सम्बन्धित नहीं है?

- (अ) FIFOसूची
- (ब) LIFOसूची
- (स) POP
- (द) PUSH

**उत्तर:** (अ) FIFOसूची

**प्रश्न 2.** शब्द Push और POP किस से सम्बन्धित है?

- (अ) ऐरे
- (ब) लिस्ट
- (स) स्टैक
- (द) ये सभी

**उत्तर:** (स) स्टैक

**प्रश्न 3.** एक डेटा स्ट्रक्चर जहाँ तत्वों का जोड़ना या हटाना किसी भी सिरे पर किया जा सकता है लेकिन बीच में नहीं।

- (अ) लिंक लिस्ट
- (ब) स्टैक
- (स) क्यू
- (द) डीक्यू

**उत्तर:** (द) डीक्यू

**प्रश्न 4.** ग्राफ में Breadth First Traversal के लिए आवश्यक डेटा स्ट्रक्चर है।

- (अ) स्टैक
- (ब) ऐरे
- (स) क्यू
- (द) ट्री (Tree)

**उत्तर:** (स) क्यू

**प्रश्न 5. एक क्यू है**

(अ) FIFO लिस्ट

(ब) LIFO लिस्ट

(स) ओर्डर्ड ऐरे

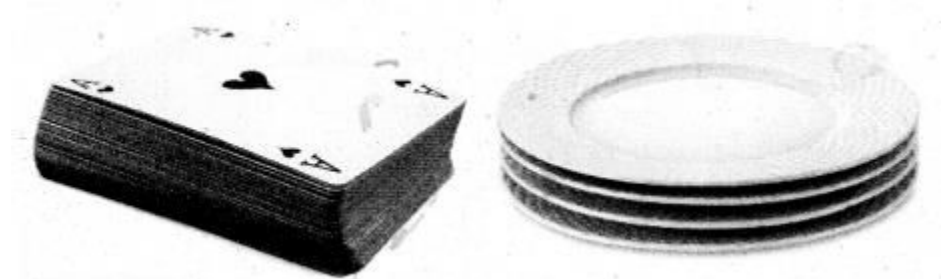
(द) रैखिक ट्री

**उत्तर:** (अ) FIFO लिस्ट

## लघु उत्तरीय प्रश्न

**प्रश्न 1. स्टैक को परिभाषित कीजिए।**

**उत्तर-** स्टैक (Stack)-स्टैक एक एब्स्ट्रैक्ट डाटा टाइप (एडीटी) है जिसका प्रयोग आमतौर पर सभी प्रोग्रामिंग भाषाओं में किया जाता है। उदाहरण के लिए ताश के पत्तों की स्टैक या एक डेक या प्लेटों की स्टैक वास्तविक दुनिया में एक स्टैक की तरह बर्ताव करती है।



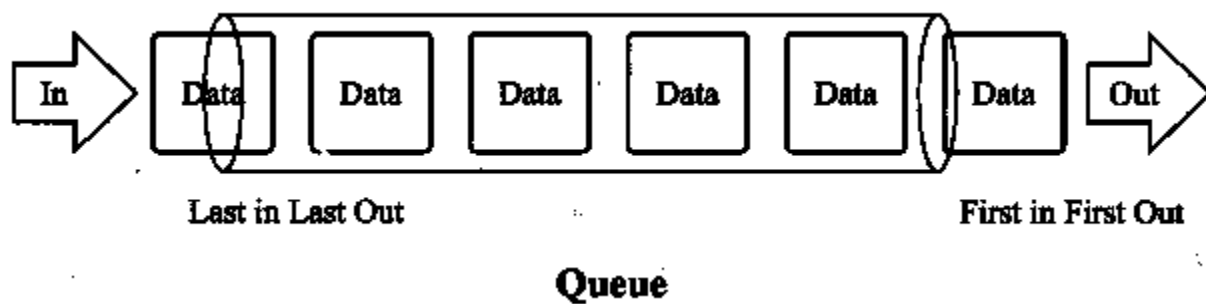
वास्तविक दुनिया में स्टैक केवल एक छोर पर ऑपरेशन की अनुमति देती है। उदाहरण के लिए-हम केवल कार्ड या प्लेट स्टैक के ऊपर से तत्व रख या निकाल सकते हैं। इसी तरह, स्टैक एडीटी केवल एक छोर पर डेटा के ऑपरेशन की अनुमति देता है। किसी भी समय, हम केवल स्टैक के शीर्ष तत्व का उपयोग कर सकते हैं। यह सुविधा स्टैक को LIFO (लास्ट-इन-फर्स्ट-आउट) डेटा स्ट्रक्चर बनाता है। यहाँ पर जो तत्व अंत में जोड़ा जाता है पहले हटाया जाता है स्टैक शब्दावली में जोड़ने को पुश (Push) तथा हटाने को पॉप (Pop) ऑपरेशन कहा जाता है।

**प्रश्न 2. क्यू को परिभाषित कीजिए।**

**उत्तर-** क्यू (Queue)-क्यू एक एब्स्ट्रैक्ट डेटा स्ट्रक्चर है जो कुछ हद तक स्टैक के समान है। स्टैक के विपरीत, एक क्यू अपने दोनों सिरों पर खुला होता है। एक सिरा तत्व को जोड़ने (enqueue) और दूसरा सिरा तत्व को हटाने (dequeue) के उपयोग में आता है। क्यू डेटा स्ट्रक्चर पहले आओ पहले जाओ (FIFO) के सिद्धान्त पर काम करता है अर्थात्, पहले संग्रहीत डेटा आइटम पहले हटा दिए जायेंगे।



क्यू के लिए एक वास्तविक दुनिया में एक सिंगल लेन का रोड जिसमें जो साधन पहले प्रवेश करेगा पहले बाहर आएगा एक उदाहरण के रूप में देखा जा सकता है।



**प्रश्न 3. पुश ऑपरेशन क्या है?**

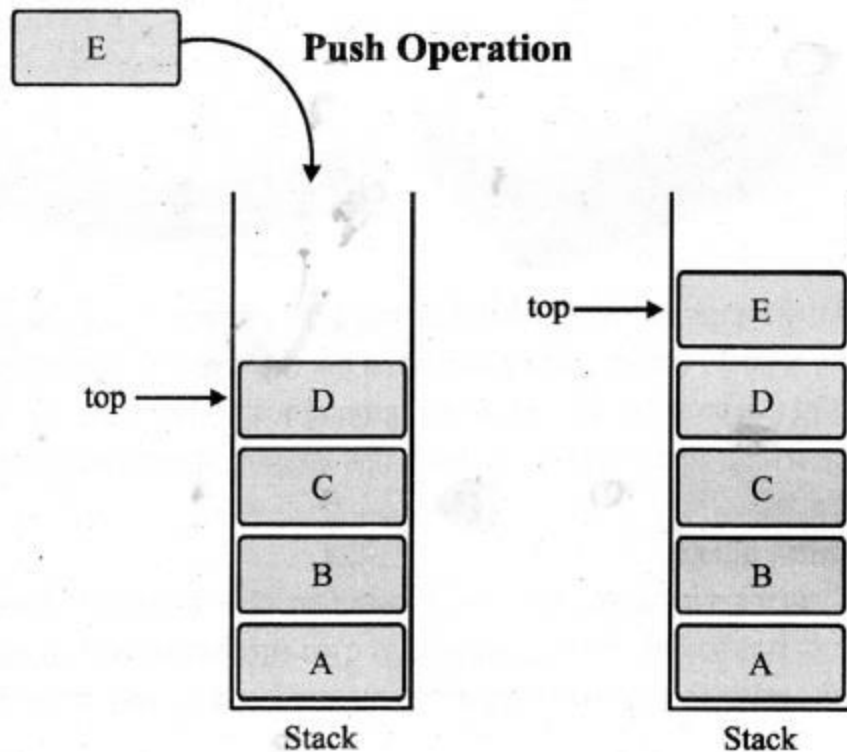
**उत्तर-** पुश ऑपरेशन (Push Operation)-स्टैक में एक नया तत्व जोड़ने या डालने की प्रक्रिया को पुश ऑपरेशन कहते हैं। पुश ऑपरेशन निम्नलिखित स्टेप्स की एक श्रृंखला है।

Step 1 – Checks if the stack is full.

Step 2 – If the stack is full, produces an error and exit.

Step 3 – If the stack is not full, increments top to point next empty space.

Step 4 – Adds data elements to the stack location, where top is pointing.



Stack Step 5 Returns success.

यदि स्टैक को लिंक लिस्ट से इम्पलीमेंट करते हैं तो स्टेप 3 में डायनामिक मेमोरी आवंटित करनी होगी।

#### प्रश्न 4. पॉप ऑपरेशन क्या है?

**उत्तर-** पॉप ऑपरेशन (Pop Operation) : स्टैक से एक डेटा तत्व को हटाने की प्रक्रिया को पॉप ऑपरेशन कहते हैं। जब पॉप ऑपरेशन को एक ऐरे की मदद से इम्पलीमेंट करते हैं तो वास्तव में डेटा तत्व को हटाने की बजाए टॉप वेरिएबल को एक से घटाते हैं। जबकि लिस्ट से इम्पलीमेंट करने पर वास्तव में डेटा तत्व को हटा कर मेमोरी को deallocated किया जाता है पॉप ऑपरेशन। निम्नलिखित स्टेप्स की एक श्रृंखला है।

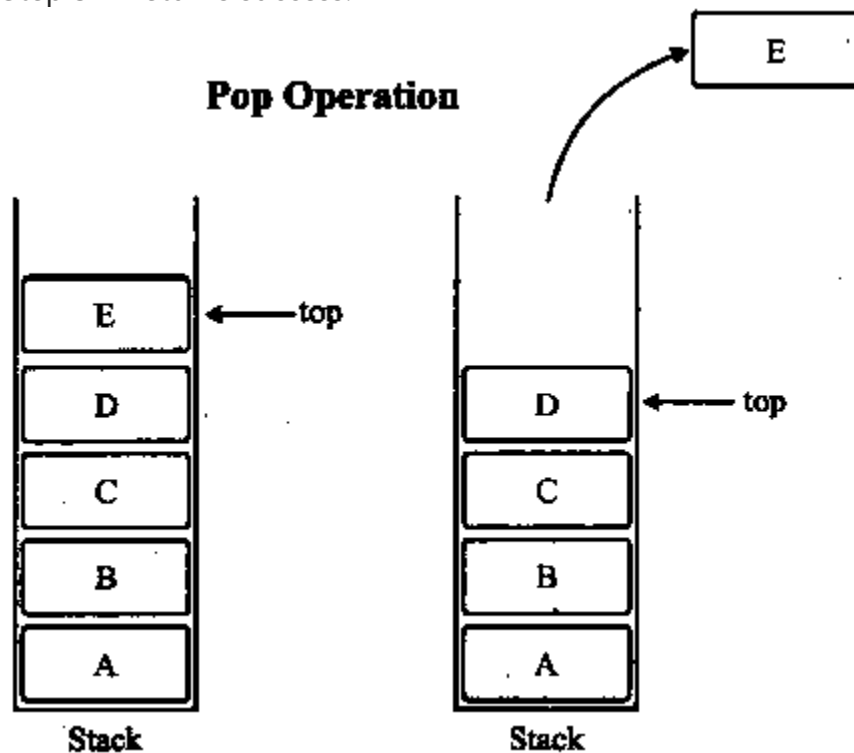
Step 1 – Checks if the stack is empty.

Step 2 – If the stack is empty, produces an error and exit.

Step 3 – If the stack is not empty, accesses the data element at which top is pointing.

Step 4 – Decreases the value of top by 1.

Step 5 – Returns success.



## निबंधात्मक प्रश्न

**प्रश्न 1. स्टैक डेटा स्ट्रक्चर के एप्लिकेशन को समझाइए।**

**उत्तर-** स्टैक डाटा स्ट्रक्चर एप्लिकेशन-निम्नलिखित कार्यों के लिये स्टैक का उपयोग किया जा सकता है  
(क) अंकगणित अभिव्यक्ति मूल्यांकन (अर्थमैटिक एक्सप्रेसन इवैल्यूएशन)  
(ख) बकट्रैकिंग  
(ग) स्मृति प्रबन्धन (मैमोरी मैनेजमेंट)

(क) अंकगणित अभिव्यक्ति मूल्यांकन (अर्थमैटिक एक्सप्रेसन इवैल्यूएशन) – अर्थमैटिक एक्सप्रेसन लिखने के तरीके को नोटेशन कहते हैं। एक अर्थमैटिक एक्सप्रेसन को तीन अलग-अलग तरीकों लेकिन समान नोटेशन में लिख सकते हैं बिना सार या आउटपुट के बदले।

निम्नलिखित तीन नोटेशन हैं-

इन्फिक्स नोटेशन

उपसर्ग (पोलिश) नोटेशन

पोस्टफिक्स (रिवर्स पोलिश) नोटेशन

इन एक्सप्रेसन का नाम ऑपरेटर के उपयोग के अनुसार दिया गया है।

इन्फिक्स नोटेशन

हम एक एक्सप्रेसन  $a-b+c$  लिखते हैं जिसमें ऑपरेटर ऑपरेंड के मध्य इस्तेमाल किया गया है। ये एक

इन्फिक्स नोटेशन है। इसका मनुष्य के लिये पढ़ना, लिखना और बोलना आसान है लेकिन कम्प्यूटिंग उपकरणों के लिए मुश्किल है एक एल्गोरिथ्म में इन्फिक्स। एक्सप्रेसन को प्रोसेस करने के लिए अधिक टाइम और स्पेस की आवश्यकता होती है।

उपसर्ग (पोलिश) नोटेशन

इस नोटेशन में ऑपरेटर ऑपरेंड के आगे लिखा होता है उदाहरण के लिए  $+ ab$  जो कि इन्फिक्स नोटेशन  $a + b$  के समान है। उपसर्ग नोटेशन को पोलिश नोटेशन भी कहते हैं।

पोस्टफिक्स नोटेशन

पोस्टफिक्स नोटेशन को रिवर्स पोलिश नोटेशन कहते हैं। इसमें ऑपरेटर ऑपरेंड के बाद में होता है उदाहरण के लिए  $ab+$  जो की इन्फिक्स नोटेशन  $a + b$  के समान है। स्टैक का उपयोग एक नोटेशन को दूसरे नोटेशन में रूपान्तरण के लिए किया जाता है।

(ख) बकट्रैकिंग : बकट्रैकिंग का प्रयोग एल्गोरिथ्म में किया जाता है जहाँ किसी पथ के साथ स्टेप्स होते हैं जो किसी स्टार्ट पॉइंट से किसी उद्देश्य तक हो। उदाहरण के लिए एक भूलभुलैया के माध्यम से अपना रास्ता सर्व करना। एक ग्राफ में एक पॉइंट से दूसरे पॉइंट तक रास्ता पता करना।

उपरोक्त सभी मामलों में एक पॉइंट से दूसरे पॉइंट तक जाने के लिए बहुत सारे विकल्प होते हैं यदि एक पॉइंट से दूसरे पॉइंट पर जाने के बाद वापस पहले पॉइंट पर आना हो और अन्य विकल्प चुनना हो।

फिर, समाधान के लिए स्टैक का इस्तेमाल किया जा सकता है। रिकर्शन एक अन्य ईस्ट समाधान है जिसको स्टैक की मदद से इम्प्लीमेंट कर सकते हैं।

(ग) स्मृति प्रबन्धन – कोई भी आधुनिक कम्प्यूटर अपने प्रोग्राम को रन करने के लिए प्राथमिक स्मृति प्रबन्धन मॉडल के रूप में एक स्टैक उपयोग करता है।

## प्रश्न 2. स्टैक ऑपरेशन को विस्तार से समझाइए।

**उत्तर-** स्टैक ऑपरेशन-स्टैक ऑपरेशन का उपयोग स्टैक को इनिशलायजिंग और डीइनिशलायजिंग करने के लिए किया जाता है। इसके अलावा एक स्टैक निम्नलिखित दो प्राथमिक कार्यों के लिए प्रयोग किया जाता है

Push () – एक तत्व स्टैक में जोड़ना

Pop () – एक तत्व स्टैक से हटाना

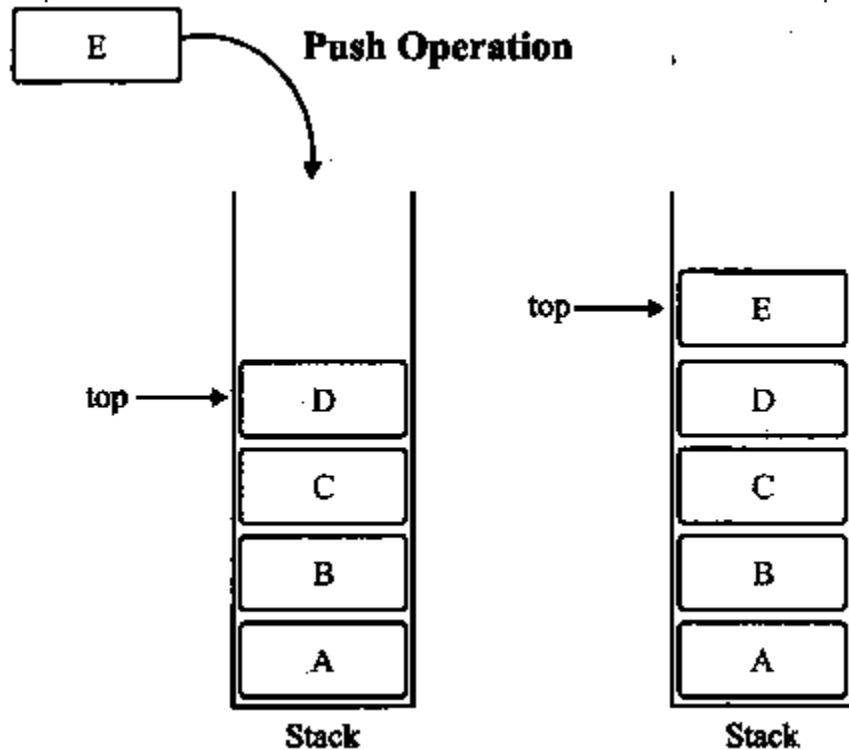
पुश ऑपरेशन (Push Operation) – स्टैक में एक नया डेटा तत्व जोड़ने या डालने की प्रक्रिया को पुश ऑपरेशन कहते हैं। पुश ऑपरेशन निम्नलिखित स्टेप्स की एक श्रृंखला है।

Step 1 – Checks if the stack is full.

Step 2 – If the stack is full, produces an error and exit.

Step 3 – If the stack is not full, increments top to point next empty space.

Step 4 – Adds data element to the stack location, where top is pointing.



Stack Step 5 – Returns success.

यदि स्टैक का लिंक लिस्ट से इम्प्लीमेंट करते है तो स्टेप 3 में डायनामिक मैमोरी आवंटित करनी होगी।

पॉप ऑपरेशन (Pop Operation) : स्टैक से एक डेटा तत्त्व को हटाने की प्रक्रिया को पॉप ऑपरेशन कहते है। जब पॉप ऑपरेशन को एक ऐरे की मदद से इम्प्लीमेंट करते है तो वास्तव में डाटा तत्त्व को हटाने की बजाए टॉप वेरिएबल को एक से घटते है जबकि लिंक लिस्ट से इम्प्लीमेंट करने पर वास्तव में डाटा तत्त्व को हटा कर मैमोरी को deallocates किया जाता है। पॉप ऑपरेशन

**निम्नलिखित स्टेप्स की एक श्रृंखला है**

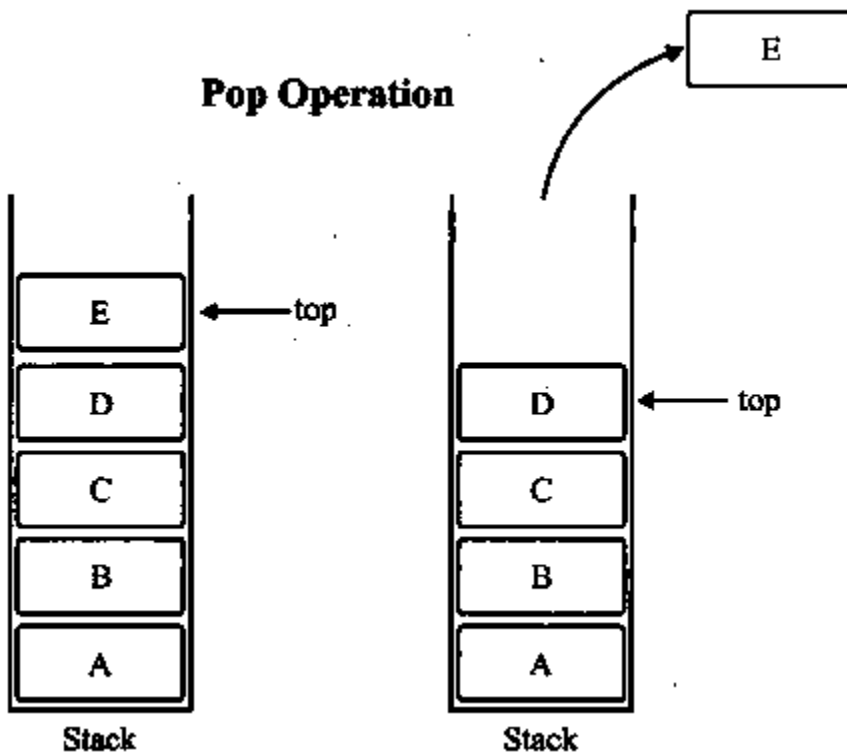
Step 1 – Checks if the stack is empty.

Step 2 – If the stack is empty, produces an error and exit.

Step 3 – If the stack is not empty, accesses the data element at which top is pointing.

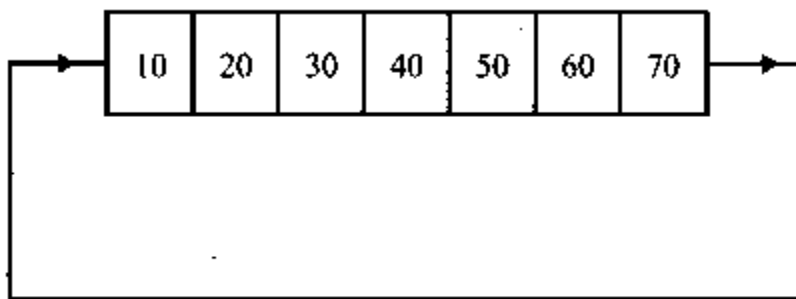
Step 4 – Decreases the value of top by 1.

Step 5 – Returns success.



प्रश्न 3. विस्तार से सर्क्युलर क्यू को समझाइए।

**उत्तर- सर्क्युलर क्यू (Circular Queue)** – सर्क्युलर क्यू को हम रिंग बफर (ring buffer) भी कहते हैं। सर्क्युलर क्यू में जो अन्तिम नोड होता है वह सबसे पहले नोड से जुड़ा हुआ रहता है जिससे की circle का निर्माण होता है। यह FIFO के सिद्धान्त पर कार्य करता है। सर्क्युलर क्यू में item को rear end से add किया जाता है तथा item को front end से remove किया जाता है।



**Circular Queue**

सर्क्युलर क्यू पर मुख्यतः दो प्रकार के ऑपरेशन्स परफॉर्म किए जा सकते हैं।

**1. इन्सर्ट ऑपरेशन (Insert Operation)** – सर्क्युलर क्यू में किसी तत्व को इन्सर्ट करने के लिए insert operations का प्रयोग किया जाता है।

एल्गोरिथ्म (Algorithm)



Step 1 – If REAR= SIZE-1 then  
REAR=0  
Else  
REAR= REAR+1  
Step 2 – If FRONT = REAR then  
write ("Circular Queue Overflow")  
Step 3 – CQ[REAR]=X  
Step 4 – If FRONT =-1 then  
FRONT=0

**2. डिलीट ऑपरेशन (Delete Operation)** – सक्यूलर क्यू में से किसी तत्व को हटाने के लिए Delete ऑपरेशन का प्रयोग किया जाता है।

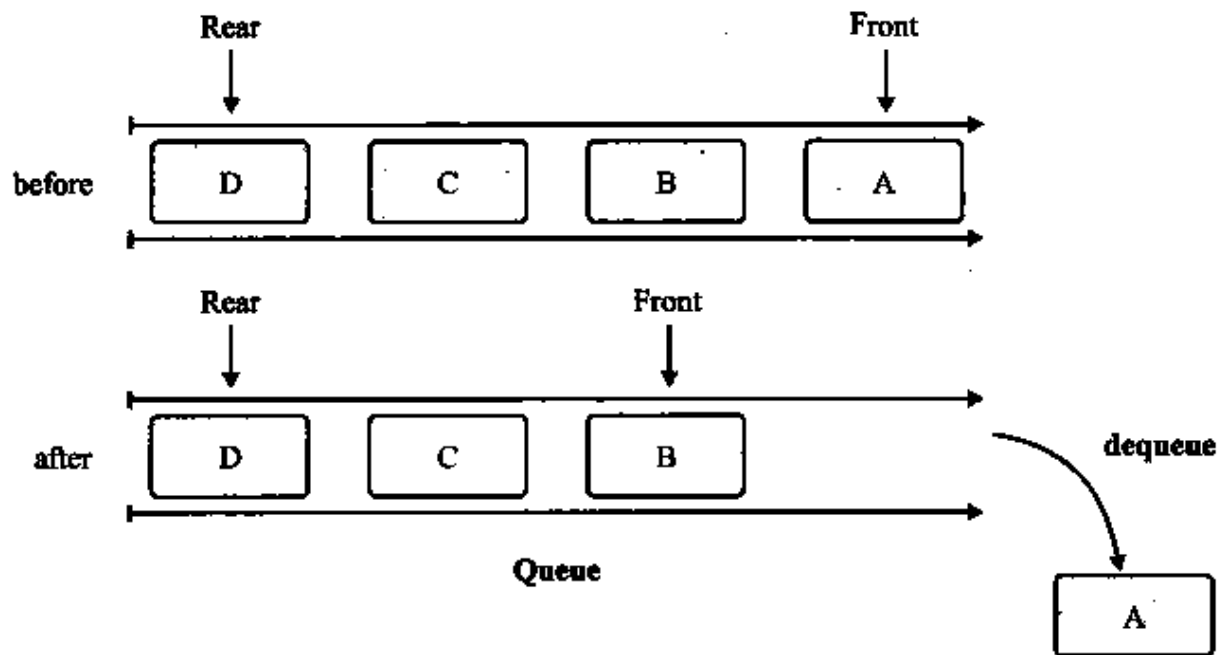
एल्गोरिथ्म (Algorithm)

Step 1 – If FRONT=-1 then  
Write ("Circular Queue Underflow")  
Step 2 – Return (CQ[FRONT])  
Step 3 – If FRONT = Size – 1 then  
FRONT=0  
Else  
FRONT=FRONT +1

#### **प्रश्न 4. डीक्यू को समझाइए।**

**उत्तर-** डीक्यू (Dequeue) ऑपरेशन-डाटा तत्व को क्यू से हटाने का काम दो भागों में किया जाता है। एक उस डाटा तत्व को एक्सेस करना जहाँ पॉइंटर पॉइंट कर रहा हो और दूसरा उसको वहाँ से हटाना। निम्नलिखित स्टेप्स से डीक्यू ऑपरेशन परफॉर्म किया जाता है

Step 1 – check if the queue is empty.  
Step 2 – If the queue is empty, produce underflow error and exit.  
Step 3 – If the queue is not empty, access the data where front is pointing.  
Step 4 – Increment front pointer to point to the next available data element.  
Step 5 – Return success.



### Queue Dequeue

डीक्यू (Dequeue) ऑपरेशन के लिए एल्गोरिथ्म

```
procedure dequeue
```

```
if queue is empty
```

```
return underflow
```

```
end if
```

```
data = queue[front]
```

```
front ← front+1
```

```
return true
```

```
end procedure
```

डीक्यू (Dequeue) ऑपरेशन का C भाषा में इम्प्लीमेंटेशन

```
int dequeue ()
```

```
{
```

```
if(isempty())
```

```
return 0;
```

```
int data = queue[front];
```

```
front = front + 1;
```

```
return data;
```

```
}
```

## अन्य महत्वपूर्ण प्रश्न

### अतिलघु उत्तरीय प्रश्न

**प्रश्न 1. एक स्टैक को कैसे इम्प्लीमेंट किया जाता है?**

**उत्तर-** एक स्टैक को ऐरे, स्ट्रक्चर, पॉइंटर और लिंकड लिस्ट के माध्यम से इम्प्लीमेंट किया जा सकता है।

**प्रश्न 2. स्टैक में `peak ()` फंक्शन का कार्य बताइए।**

**उत्तर-** `peak()` फंक्शन का कार्य स्टैक के शीर्ष डेटा तत्व को हटाये बिना प्राप्त करना है।

**प्रश्न 3. स्टैक में `is full()` फंक्शन का क्या कार्य है?**

**उत्तर-** `isfull ()` फंक्शन स्टैक के भरे होने की जाँच करता है।

**प्रश्न 4. स्टैक में `isEmpty ()` फंक्शन का क्या कार्य है?**

**उत्तर-** स्टैक में `isEmpty ()` फंक्शन स्टैक के खाली होने की जाँच करता है।

**प्रश्न 5. पॉप (Pop) ऑपरेशन को एक ऐरे की मदद से इम्प्लीमेंट करने पर क्या होता है?**

**उत्तर-** जब पॉप (Pop) ऑपरेशन को एक ऐरे की मदद से इम्प्लीमेंट करते हैं, तो वास्तव में डाटा तत्व को हटाने की बजाए टॉप वेरिएबल को एक से घटाते हैं।

**प्रश्न 6. पॉप (Pop) ऑपरेशन को लिंक लिस्ट में इम्प्लीमेंट करने पर क्या होता है?**

**उत्तर-** जब पॉप (Pop) ऑपरेशन को लिंक लिस्ट से इम्प्लीमेंट करते हैं तो वास्तव में डाटा तत्व को हटा कर मेमोरी को deallocate किया जाता है।

**प्रश्न 7. स्टैक ऑपरेशन का मुख्य उपयोग बताइए।**

**उत्तर-** स्टैक ऑपरेशन का उपयोग स्टैक को इनिशलायजिंग और डीइनिशलायजिंग करने के लिए किया जाता है।

**प्रश्न 8. नोटेशन किसे कहते हैं?**

**उत्तर-** अर्थमैटिक एक्सप्रेसन लिखने के तरीके को नोटेशन कहते हैं।

**प्रश्न 9. क्या क्यू को इम्प्लीमेंट किया जा सकता है?**

**उत्तर-** एक स्टैक की तरह क्यू को भी ऐरे, स्ट्रक्चर, पॉइंटर और लिंकड लिस्ट के माध्यम में इम्प्लीमेंट किया जा सकता है।

**प्रश्न 10. क्यू (Queue) में कितने डाटा पॉइंटर होते हैं?**

**उत्तर-** क्यू (Queue) में दो डाटा पॉइंटर होते हैं।

- फ्रंट (Front)
- रियर (Rear)

## लघु उत्तरीय प्रश्न

**प्रश्न 1. peek () फंक्शन के लिए एल्गोरिथ्म और C भाषा में इम्प्लीमेंटेशन लिखिए।**

**उत्तर-** peek () फंक्शन के लिए एल्गोरिथ्म  
begin procedure peek  
return stack[top]  
end procedure  
peek() फंक्शन का C भाषा में इम्प्लीमेंटेशन  
int peek()  
{  
return stack[top];  
}

**प्रश्न 2. isfull () फंक्शन के लिए एल्गोरिथ्म और C भाषा में इम्प्लीमेंटेशन लिखिए।**

**उत्तर-** isfull () फंक्शन के लिए एल्गोरिथ्म  
begin procedure isfull  
if top equals to MAXSIZE  
return true  
else  
return false  
endif  
end procedure  
isfull() फंक्शन का C भाषा में इम्प्लीमेंटेशन

```

bool isfull()
ifr(top== MAXSIZE)
return true;
else
return false;
}

```

**प्रश्न 3. isempty () फंक्शन के लिए एल्गोरिथ्म और C भाषा में इम्प्लीमेंटेशन लिखिए।**

**उत्तर-** isempty () फंक्शन के लिए एल्गोरिथ्म

```

begin procedure isempty
if top less than 1
return true
else
return false
endif
end procedure

```

isempty() फंक्शन का C भाषा में इम्प्लीमेंटेशन

Example

```

bool isempty()
{
if(top== -1)
return true;
else
return false;
}

```

**प्रश्न 4. पुश (Push) ऑपरेशन के लिए एल्गोरिथ्म और C में इम्प्लीमेंटेशन लिखिए।**

**उत्तर-** पुश ऑपरेशन के लिए एल्गोरिथ्म

```

begin procedure push : stack, data
if stack is full
return null
endif
top ← top + 1
stack[top] ← data
end procedure

```

एल्गोरिथ्म का C में इम्प्लीमेंटेशन

```

void push(int data)

```

```

{
if(!isFull())
{
top = top + 1;
stack[top] = data;
}
else
{
printf("Could not insert data, Stack is full.\n");
}
}

```

**प्रश्न 5. पॉप (Pop) ऑपरेशन के लिए एल्गोरिथ्म और C में इम्प्लीमेंटेशन लिखिए।**

**उत्तर-** पॉप ऑपरेशन के लिए एल्गोरिथ्म

begin procedure push : stack

if stack is empty

return null

endif

data ← stack[top]

top ← top – 1

return data

end procedure

एल्गोरिथ्म को C का इम्प्लीमेंटेशन

int pop(int data)

```
{
```

```
if(!isempty())
```

```
{
```

```
data = stack[top];
```

```
top = top – 1;
```

```
return data;
```

```
}
```

```
else
```

```
{
```

```
printf("Could not retrieve data, Stack is empty.\n");
```

```
}
```

```
}
```

**प्रश्न 6. इन्फिक्स नोटेशन के विषय में बताइए।**

**उत्तर-** इन्फिक्स नोटेशन

हम एक एक्सप्रेशन  $a-b + c$  लिखते हैं जिसमें ऑपरेटर ऑपरेंड के मध्य इस्तेमाल किया गया है। ये एक इन्फिक्स नोटेशन है। इसका मनुष्य के लिये पढ़ना, लिखना और बोलना आसान है लेकिन कम्प्यूटिंग उपकरणों के लिए मुश्किल है। एक एल्गोरिथ्म में इन्फिक्स एक्सप्रेशन को प्रोसेस करने के लिए अधिक टाइम और स्पेस की आवश्यकता होती है।

**प्रश्न 7. उपसर्ग (पोलिश) नोटेशन किसे कहते हैं?**

**उत्तर-** उपसर्ग (पोलिश) नोटेशन-इस नोटेशन में ऑपरेटर ऑपरेंड के आगे लिखा होता है उदाहरण के लिए  $+ab$  जो कि इन्फिक्स नोटेशन  $a + b$  के समान है उपसर्ग नोटेशन को पोलिश नोटेशन भी कहते हैं।

**प्रश्न 8. पोस्टफिक्स नोटेशन किसे कहते हैं?**

**उत्तर-** पोस्टफिक्स नोटेशन-पोस्टफिक्स नोटेशन को रिवर्स पोलिश नोटेशन कहते हैं इसमें ऑपरेटर ऑपरेंड के बाद में होता है उदाहरण के लिए  $ab+$  जो की इन्फिक्स नोटेशन  $a + b$  के समान है।

## निबंधात्मक प्रश्न

**प्रश्न 1. क्यू (Queue) के कुशलता से उपयोग के लिए प्रयुक्त फंक्शन बताइए।**

अथवा

**क्यू (Queue) में उपयोग होने वाले प्रमुख फंक्शन्स के विषय में बताइए।**

**उत्तर-** क्यू के कुशलता से उपयोग के लिए निम्नलिखित फंक्शनस का उपयोग करते हैं

Peek() – क्यू के शीर्ष डेटा तत्त्व को हटये बिना प्राप्त करना,

isfull() – क्यू के भरे होने की जाँच करना,

isempty () – क्यू के खाली होने की जाँच करना

क्यू के सपोर्टिव फंक्शनस निम्न है।

peek ()

peek() फंक्शन के लिए एल्गोरिथ्म –

begin procedure peek

return queue[front]

end procedure

peek() फंक्शन का C भाषा में इम्प्लीमेंटेशन –

Example

```

int peek()
{
return queue[front];
isfull() फंक्शन के लिए एल्गोरिथ्म
begin procedure isfull
if rear equals to MAXSIZE
return true
else
return false
endif
end procedure

```

isfull() फंक्शन का C भाषा में इम्प्लीमेंटेशन-

```

bool isfull()
{
if(rear== MAXSIZE-1)
return true;
else
return false;
}
isempty () :

```

isempty () फंक्शन के लिए एल्गोरिथ्म-

```

being procedure isempty
if front is less than MIN OR front is greater than rear
return true
else
return false
endif
end procedure

```

यदि फ्रंट का मान 0 से कम है तो इसका मतलब क्यू को इनिशलायज नहीं किया है और क्यू खाली है।

isempty() फंक्शन को C भाषा में इम्प्लीमेंटेशन-

```

bool isempty()
{
if(front<0|| front>rear)
return true;
else
return false;
}

```



**प्रश्न 2. क्यू (Queue) के बुनियादी ऑपरेशन कौन से हैं? किसी एक का वर्णन कीजिए।**

**अथवा**

**enqueue () ऑपरेशन के विषय में विस्तार से बताइए।**

**उत्तर-** क्यू के बुनियादी ऑपरेशन : क्यू ऑपरेशन का उपयोग क्यू को इनिशलायजिंग और डीइनिशलायजिंग करने के लिए किया जाता है। निम्नलिखित क्यू के बुनियादी ऑपरेशन हैं।

enqueue () – एक तत्व क्यू में जोड़ना

dequeue () – एक तत्व क्यू से हटाना

एनक्यू (Enqueue) ऑपरेशन – क्यू में दो डाटा पॉइंटर फ्रंट और रियर होते हैं इसलिए इसके ऑपरेशन स्टैक से कठिन होते हैं। क्यू में डाटा तत्व को जोड़ने (insert) के लिए निम्नलिखित स्टेप्स का उपयोग करते हैं।

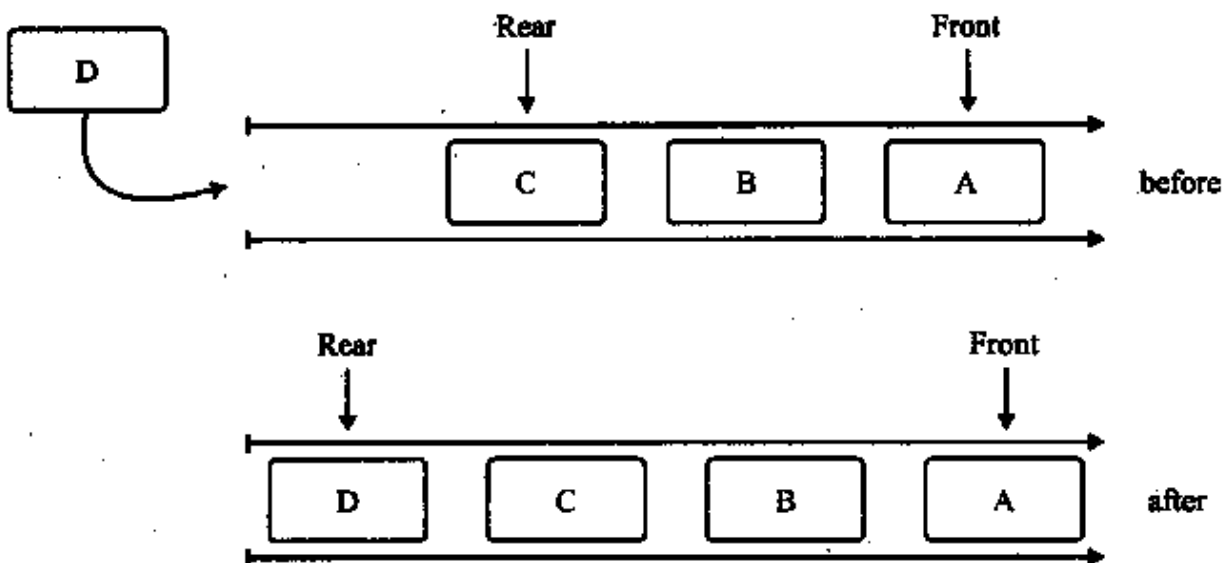
Step 1 – Check if the queue is full.

Step 2 – If the queue is full, produce overflow error and exit.

Step 3 – If the queue is not full, increment rear pointer to point the next empty space.

Step 4 – Add data element to the queue location, where the rear is pointing.

Step 5 – return success.



### **Queue Enqueue**

एनक्यू (Enqueue) ऑपरेशन के लिए एल्गोरिथ्म

procedure enqueue (data)

if queue is full

```
return overflow
endif
rear←rear + 1
queue[rear]←data
return true
end procedure
```

एनक्वू (Enqueue) ऑपरेशन की C भाषा में इम्प्लीमेंटेशन

```
int enqueue (int data)
if(isfull())
return 0;
rear=rear + 1;
queue[rear] = data;
return 1;
end procedure
```