

Subject: \_\_\_\_\_

Date: 

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

  
Sat ☐ Sun ☐ Mon ☐ Tue ☐ Wed ☐ Thu ☐ Fri ☐

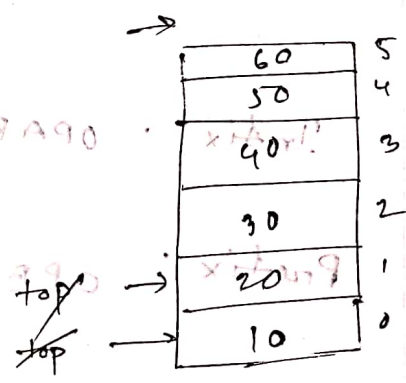
## Stack

// last = last

1. push() → Insert  
→ To get out at last value what I insert
2. pop() → To remove
3. peek() → which I can insert first just see
4. isEmpty() → If I remove anything then it say false
5. Stack overflow Exception

push (10)

- (20)
- (30)
- (40)
- (50)
- (60)
- (70)
- (80)



(90) → stack overflow Exception

pop (50)

- (40)
- (30)
- (20)
- (10)

(0) → stack underflow Exception

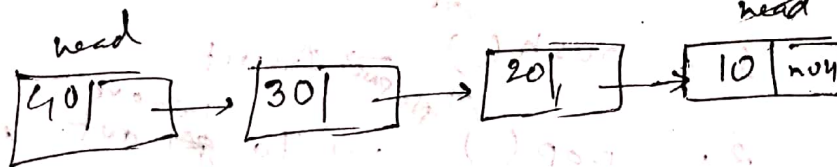
Subject:

Date:

Sat ☐ Sun ☐ Mon ☐ Tue ☐ Wed ☐ Thu ☐ Fri ☐

head = null

push (20)  
(30)  
(40)



In linked list there will be no space of stack  
overflow Exception.

Infix : OPERAND1

OPERATOR

OPERAND2

Prefix : OPERATOR

OPERAND1

OPERAND2

Postfix : OPERAND1

OPERAND2

OPERATOR

\* OPERATOR Precedences

( )

{ }

[ ]

x ^ / %

+ -

For same priority left to right.

①

Infix :  $A - B / (C * D^E)$

$$\downarrow$$

$$(C * D^E)$$

$$C D E^{\wedge} *$$

$$B C D E^{\wedge} * /$$

Postfix :  $A B C D E^{\wedge} * / -$

Prefix :  $E^{\wedge} D C *$

$$B / * C^{\wedge} D E$$

$$A - / B * C^{\wedge} D E$$

$$- A / B * C^{\wedge} D E$$

②

Infix :  $2 * 3 / (2 - 1) + 5 * 3$

Postfix :  $2 * 3 / 2 1 - + 5 * 3$

$$2 * 3 2 1 / - + 5 3 *$$

$$2 3 * / 2 1 - + 5 3 *$$

$$2 3 * 2 1 - / 5 3 * +$$

$$2 3 2 1 * / - 5 3 -$$

$$2 3 * 2 1 - / 5 3 * +$$

$$2 3 * 2 1 / -$$

Prefix :  $2 * 3 / - 2 1 + 5 * 3$

$$* 2 3 / - 2 1 + * 5 3$$

$$/ * 2 3 - 2 1 + 2 1 -$$

$$+ / * 2 3 - 2 1 * 5 3$$

(iii)

Infix :  $(A + B / C * (D + E) - F)$ Postfix :  $(A + B / C * DE + - F)$  $(A + BC / * DE + - F)$  $A + BC / DE * + - F$  $ABC / DE + * + F -$ Infix to Postfix using stack rule's

1st rule : ~~Know~~ to No same operators of the same priority can stay together.

The operator that was already in the stack will be popped out.

2nd rule : ~~Know~~ No lower priority operators can stay on top of a higher priority operator.

The higher priority operator will be popped out.

3rd rule : Whenever a closing bracket is found all the operators in between the braces/bracket will be popped out from right to left.



Subject: \_\_\_\_\_

Date: 

|  |  |  |
|--|--|--|
|  |  |  |
|--|--|--|

  
Sat ☐ Sun ☐ Mon ☐ Tue ☐ Wed ☐ Thu ☐ Fri ☐

9th rule: At the end, if any operator left in the stack there will be popped from right to left.

eg:  $(A + B / C * (D + E) - F)$

| Expression | stack(operator)                                    | Operand + popped values |
|------------|--|-------------------------|
| (          | (  | A                       |
| A          |  |                         |
| +          | ( +  | AB                      |
| B          |  |                         |
| /          | ( + /  | ABC                     |
| C          |  |                         |
| *          | ( + *<br><del>( +</del>                            | ABC /                   |
| (          | ( + * (  |                         |
| D          |  | ABC / D                 |
| +          | ( + * ( +  |                         |
| E          | ( + * <del>( +</del> <sup>popped</sup>             | ABC / DE +              |
| )          | ( <del>+</del> <del>*</del> <sup>POP POP</sup> - ) | ABC / DE + * +          |
| -          | ( <del>+</del> <del>*</del> <del>-</del> )         |                         |
| F          | ( <del>+</del> <del>*</del> <del>-</del> )         | ABC / DE + * + F        |
| )          | ( <del>+</del> <del>*</del> <del>-</del> )         | ABC / DE + * + F -      |

Date:

Subject:

Sat ☐ Sun ☐ Mon ☐ Tue ☐ Wed ☐ Thu ☐ Fri ☐

( ) { } [ ]

1. \* / %

2. + -

3. < <= > >=

4. == !=

5. &&

6. ||

priority decreasing

[ (4 \* 2 > 9 / 3 + 5 && 4 == 6 - 2) != 5 < 5 ] == 6 > 7 % 1

Expression

stack (operation)

operand + popped value

[

[

4

(

[(

4 2

4

[( \*

4 2 \*

\*

[( \* )   
 popped

4 2 \* 9

2

[( \* )

4 2 \* 9 3

>

[( )

4 2 \* 9 3 1

9

[( > )

4 2 \* 9 3 1 5

/

[( > ) +   
 popped

4 2 \* 9 3 1 5 +

3

[( > ) +

+

[( > ) + &&   
 popped

5

[( &&

&&

Subject: ✓

10/02/2020

Date:         
Sat ☐ Sun ☐ Mon ☐ Tue ☐ Wed ☐ Thu ☐ Fri

4

[ ( & &

42 \* 93 \ 5 + > 4

= =

[ ( & & = =

42 \* 93 \ 5 + > 462

6

[ ( & & = -

42 \* 93 \ 5 + > 462 = 62

-

[ ( & & = = -

42 \* 93 \ 5 + > 462 = = & &

2

[ ( & & = = -

Final answer

42 \* 93 \ 5 + > 462 = =

)

[ ! =

& & 55 < ! = 67 1 % > = =

! =

[ ! =

42 \* 93 \ 5 + > 462 = = & &

5

[ ! = <

42 \* 93 \ 5 + > 462 = = & & 5

<

[ ! = < ]

42 \* 93 \ 5 + > 462 = = & & 55 < ! =

5

42 \* 93 \ 5 + > 462 = = & & 55 < ! = 6

7

= =

42 \* 93 \ 5 + > 462 = = & & 55 < ! = 6

= =

= = >

6

42 \* 93 \ 5 + > 462 = = & & 55 < ! = 67

>

= = > %

42 \* 93 \ 5 + > 462 = = & & 55 < ! = 67 1

7

42 \* 93 \ 5 + > 462 = = & & 55

%

< ! = 67 1 % > = =

1