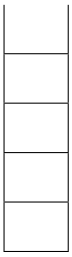
GIT Department Of Computer Engineering
CSE 222/505 - Spring 2020
Homework 4 Question 1

Fatih Kaan Salgır
171044009

1

1.1 Infix to Postfix

1. A + ((B - C * D) / E) + F - G / H

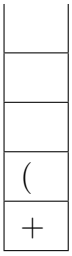
   Output: A

2. A + ((B - C * D) / E) + F - G / H

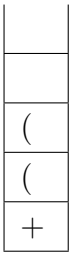
   Output: A

3. A + ((B - C * D) / E) + F - G / H

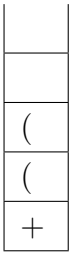
   Output: A

4. A + ((B - C * D) / E) + F - G / H

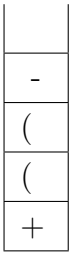
   Output: A

5. A + ((B - C * D) / E) + F - G / H

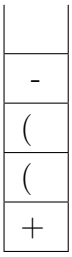
   Output: A B

6. A + ((B - C * D) / E) + F - G / H



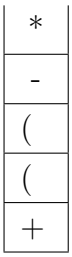
Output: A B

7. A + ((B - C * D) / E) + F - G / H



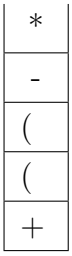
Output: A B C

8. A + ((B - C * D) / E) + F - G / H



Output: A B C

9. A + ((B - C * D) / E) + F - G / H

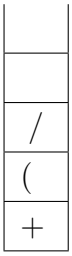


Output: A B C D

10. A + ((B - C * D) / E) + F - G / H

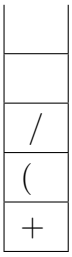
Output: A B C D * -

11. A + ((B - C * D) / E) + F - G / H



Output: A B C D * -

12. A + ((B - C * D) / E) + F - G / H



Output: A B C D * - E

13. A + ((B - C * D) / E) + F - G / H

Output: A B C D * - E /

14. A + ((B - C * D) / E) + F - G / H

Output: A B C D * - E / +

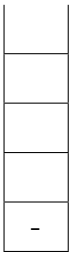
15. A + ((B - C * D) / E) + F - G / H

Output: A B C D * - E / + F

16. A + ((B - C * D) / E) + F - G / H

Output: A B C D * - E / + F +

17. A + ((B - C * D) / E) + F - G / H



Output: A B C D * - E / + F + G

18. A + ((B - C * D) / E) + F - G / H

Output: A B C D * - E / + F + G

19. A + ((B - C * D) / E) + F - G / H

Output: A B C D * - E / + F + G H

20. A + ((B - C * D) / E) + F - G / H

Output: A B C D * - E / + F + G H / -

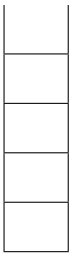
Evaluation

Output of conversion: A B C D * - E / + F + G H / -
After values assigned: 3 4 1 2 * - 2 / + 5 + 6 3 / -

Expression	Stack	Explantion
3	3	Push operand
4	3 4	Push operand
1	3 4 1	Push operand
2	3 4 1 2	Push operand
*	3 4	$1 \times 2 = 2$
	3 4 2	Push calculation result
-	3	$4 - 2 = 2$
	3 2	Push calculation result
2	3 2 2	Push operand
/	3	$2 \div 2 = 1$
	3 1	Push calculation result
+		$3 + 1 = 4$
	4	Push calculation result
5	4 5	Push operand
+		$4 + 5 = 9$
	9	Push calculation result
6	9 6	Push operand
3	9 6 3	Push operand
/		$6 \div 3 = 2$
	9 2	Push calculation result
-	9 2	$9 - 2 = 7$
	7	Push calculation result

1.2 Infix to Prefix

1. A + ((B - C * D) / E) + F - G / H

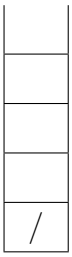


Output: H

2. A + ((B - C * D) / E) + F - G / H

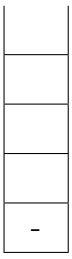
Output: H

3. A + ((B - C * D) / E) + F - G / H



Output: H G

4. A + ((B - C * D) / E) + F - G / H



Output: H G /

5. A + ((B - C * D) / E) + F - G / H

Output: H G / F

6. A + ((B - C * D) / E) + F - G / H

Output: H G / F

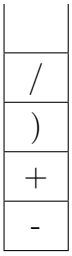
7. A + ((B - C * D) / E) + F - G / H

Output: H G / F

8. A + ((B - C * D) / E) + F - G / H

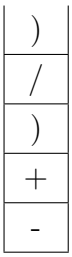
Output: H G / F E

9. A + ((B - C * D) / E) + F - G / H



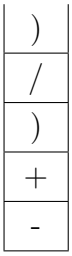
Output: H G / F E

10. A + ((B - C * D) / E) + F - G / H



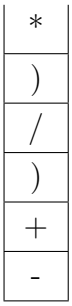
Output: H G / F E

11. A + ((B - C * D) / E) + F - G / H



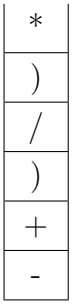
Output: H G / F E D

12. A + ((B - C * D) / E) + F - G / H



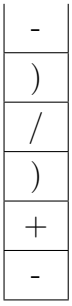
Output: H G / F E D

13. A + ((B - C * D) / E) + F - G / H



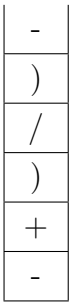
Output: H G / F E D C

14. A + ((B - C * D) / E) + F - G / H



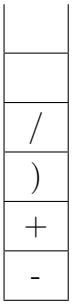
Output: H G / F E D C *

15. A + ((B - C * D) / E) + F - G / H



Output: H G / F E D C * B

16. A + ((B - C * D) / E) + F - G / H

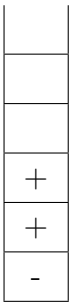


Output: H G / F E D C * B -

17. A + ((B - C * D) / E) + F - G / H

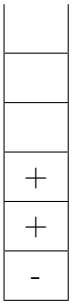
Output: H G / F E D C * B - /

18. A + ((B - C * D) / E) + F - G / H



Output: H G / F E D C * B - /

19. A + ((B - C * D) / E) + F - G / H



Output: H G / F E D C * B - / A

20. A + ((B - C * D) / E) + F - G / H

Output: H G / F E D C * B - / A + + -

After reversing we get;

Output: - + + A / - B * C D E F / G H

Evaluation

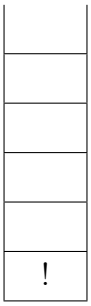
Output: H G / F E D C * B - / A + + -
After values assigned: 3 6 / 5 2 2 1 * 4 - / 3 + + -

Expression	Stack after the step	Explantion
3	3	Push operand
6	3 6	Push operand
/	2	$6 \div 3 = 2$
5	2 5	Push operand
2	2 5 2	Push operand
2	2 5 2 2	Push operand
1	2 5 2 2 1	Push operand
*	2 5 2 2	$2 \times 1 = 4$
4	2 5 2 2 4	Push operand
-	2 5 2 2	$4 - 2 = 2$
/	2 5 1	$2 \div 2 = 1$
3	2 5 1 3	Push operand
+	2 5 4	$4 + 1 = 4$
+	2 9	$5 + 1 = 9$
-	7	$9 - 2 = 7$

2

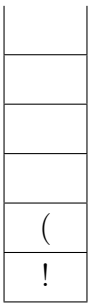
2.1 Infix to Postfix

1. ! (A && ! ((B < C) || (C > D))) || (C < E)



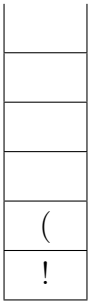
Output:

2. ! (A && ! ((B < C) || (C > D))) || (C < E)



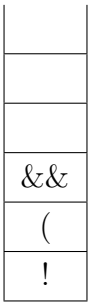
Output:

3. ! (A && ! ((B < C) || (C > D))) || (C < E)



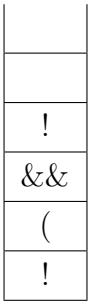
Output: A

4. ! (A && ! ((B < C) || (C > D))) || (C < E)



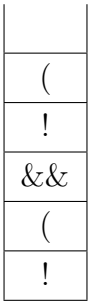
Output: A

5. ! (A && ! ((B < C) || (C > D))) || (C < E)



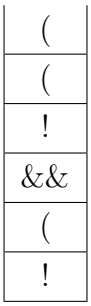
Output: A

6. ! (A && ! ((B < C) || (C > D))) || (C < E)



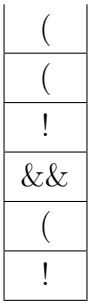
Output: A

7. ! (A && ! ((B < C) || (C > D))) || (C < E)



Output: A

8. ! (A && ! ((B < C) || (C > D))) || (C < E)



Output: A B

9. ! (A && ! ((B< C) || (C > D))) || (C < E)

<
(
(
!
&&
(
!

Output: A B

10. ! (A && ! ((B < C) || (C > D))) || (C < E)

<
(
(
!
&&
(
!

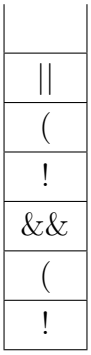
Output: A B C

11. ! (A && ! ((B < C) || (C > D))) || (C < E)

(
!
&&
(
!

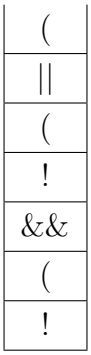
Output: A B C <

12. ! (A && ! ((B < C) || (C > D))) || (C < E)



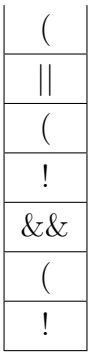
Output: A B C <

13. ! (A && ! ((B < C) || (C > D))) || (C < E)



Output: A B C <

14. ! (A && ! ((B < C) || (C > D))) || (C < E)



Output: A B C < C

15. ! (A && ! ((B < C) || (C > D))) || (C < E)

>
(
||
(
!
&&
(
!

Output: A B C < C

16. ! (A && ! ((B < C) || (C > D))) || (C < E)

>
(
||
(
!
&&
(
!

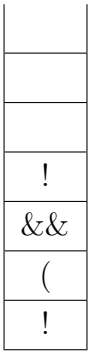
Output: A B C < C D

17. ! (A && ! ((B < C) || (C > D))) || (C < E)

||
(
!
&&
(
!

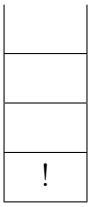
Output: A B C < C D >

18. ! (A && ! ((B < C) || (C > D))) || (C < E)



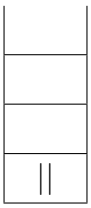
Output: A B C < C D > ||

19. ! (A && ! ((B < C) || (C > D))) || (C < E)



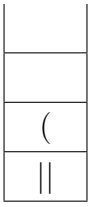
Output: A B C < C D > || ! &&

20. ! (A && ! ((B < C) || (C > D))) || (C < E)



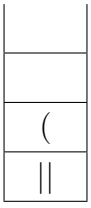
Output: A B C < C D > || ! && !

21. ! (A && ! ((B < C) || (C > D))) || (C < E)



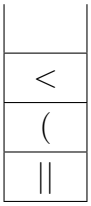
Output: A B C < C D > || ! && !

22. ! (A && ! ((B < C) || (C > D))) || (C < E)



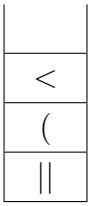
Output: A B C < C D > || ! && ! C

23. ! (A && ! ((B < C) || (C > D))) || (C < E)



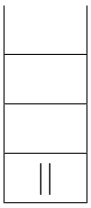
Output: A B C < C D > || ! && ! C

24. ! (A && ! ((B < C) || (C > D))) || (C < E)



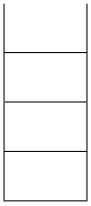
Output: A B C < C D > || ! && ! C E

25. ! (A && ! ((B < C) || (C > D))) || (C < E)



Output: A B C < C D > || ! && ! C E <

26. ! (A && ! ((B < C) || (C > D))) || (C < E)



Output: A B C < C D > || ! && ! C E < ||

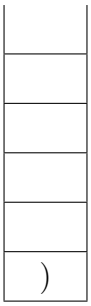
Evaluation

Output: A B C < C D > || ! && ! C E < ||
After values assigned: 1 2 3 < 3 4 > || ! && ! 3 5 < ||

Expression	Stack after the step	Explantion
1	1	Push operand
2	1 2	Push operand
3	1 2 3	Push operand
<	1 1	2 < 3 is true
3	1 1 3	Push operand
4	1 1 3 4	Push operand
>	1 1 0	3 > 4 is false
||	1 1	1 || 0 is 1
!	1 0	!1 is 0
&&	0	1 && 0 is 0
!	1	!0 is 1
3	1 3	Push operand
5	1 3 5	Push operand
<	1 1	3 < 5 is true
||	1	1 || 1 is true

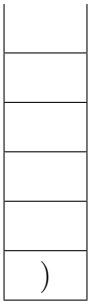
2.2 Infix to Prefix

1. ! (A && ! ((B < C) || (C > D))) || (C < E)



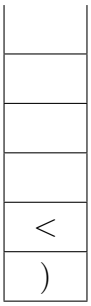
Output:

2. ! (A && ! ((B < C) || (C > D))) || (C < E)



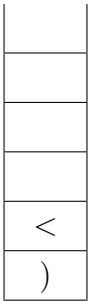
Output: E

3. ! (A && ! ((B < C) || (C > D))) || (C < E)



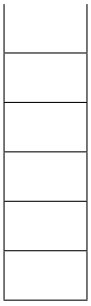
Output: E

4. ! (A && ! ((B < C) || (C > D))) || (C < E)



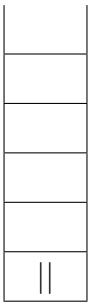
Output: E C

5. ! (A && ! ((B < C) || (C > D))) || (C < E)



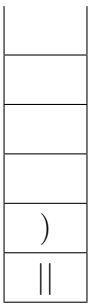
Output: E C <

6. ! (A && ! ((B < C) || (C > D))) || (C < E)



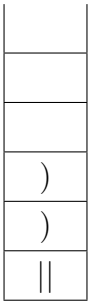
Output: E C <

7. ! (A && ! ((B < C) || (C > D))) || (C < E)



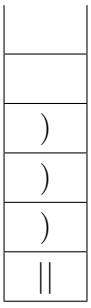
Output: E C <

8. ! (A && ! ((B < C) || (C > D))) || (C < E)



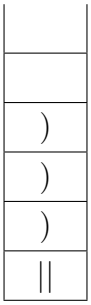
Output: E C <

9. ! (A && ! ((B < C) || (C > D))) || (C < E)



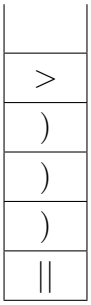
Output: E C <

10. ! (A && ! ((B < C) || (C > D))) || (C < E)



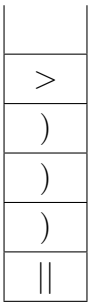
Output: E C < D

11. ! (A && ! ((B < C) || (C > D))) || (C < E)



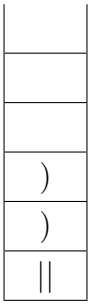
Output: E C < D

12. ! (A && ! ((B < C) || (C > D))) || (C < E)



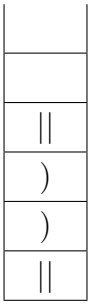
Output: E C < D C

13. ! (A && ! ((B < C) || (C > D))) || (C < E)



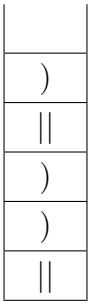
Output: E C < D C >

14. ! (A && ! ((B < C) || (C > D))) || (C < E)



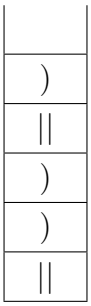
Output: E C < D C >

15. ! (A && ! ((B < C) || (C > D))) || (C < E)



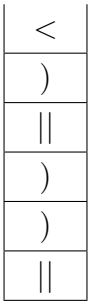
Output: E C < D C >

16. ! (A && ! ((B < C) || (C > D))) || (C < E)



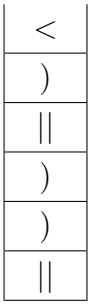
Output: E C < D C > C

17. ! (A && ! ((B < C) || (C > D))) || (C < E)



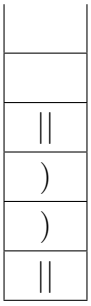
Output: E C < D C > C

18. ! (A && ! ((B < C) || (C > D))) || (C < E)



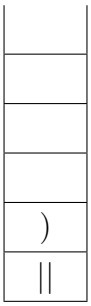
Output: E C < D C > C B

19. ! (A && ! ((B < C) || (C > D))) || (C < E)



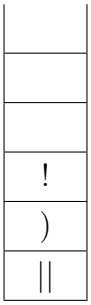
Output: E C < D C > C B <

20. ! (A && ! ((B < C) || (C > D))) || (C < E)



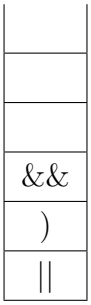
Output: E C < D C > C B < ||

21. ! (A && ! ((B < C) || (C > D))) || (C < E)



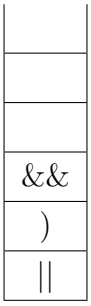
Output: E C < D C > C B < ||

22. ! (A && ! ((B < C) || (C > D))) || (C < E)



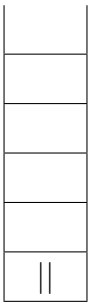
Output: E C < D C > C B < || !

23. ! (A && ! ((B < C) || (C > D))) || (C < E)



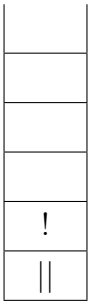
Output: E C < D C > C B < || ! A

24. ! (A && ! ((B < C) || (C > D))) || (C < E)



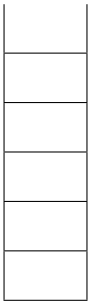
Output: E C < D C > C B < || ! A &&

25. ! (A && ! ((B < C) || (C > D))) || (C < E)



Output: E C < D C > C B < || ! A &&

26. ! (A && ! ((B < C) || (C > D))) || (C < E)



Output: E C < D C > C B < || ! A && ! ||

After reversing we get;

Output: || ! && A ! || < B C > C D < C E

Evaluation

Output: E C < D C > C B < || ! A && ! ||
After values assigned: 5 3 < 4 3 > 3 2 < || ! 1 && ! ||

Expression	Stack after the step	Explantion
5	5	Push operand
3	5 3	Push operand
<	1	3 < 5 is true
4	1 4	Push operand
3	1 4 3	Push operand
>	1 0	3 > 4 is false
3	1 0 3	Push operand
2	1 0 3 2	Push operand
<	1 0 1	2 < 3 is true
||	1 1	0 || 1 is true
!	1 0	!1 is 0
1	1 0 1	Push operand
&&	1 0	0 && 1 is false
!	1 1	!0 is 1
||	1	1 || 1 is true