

CSG2A3

ALGORITMA dan STRUKTUR DATA



Stack

Exercise

Exercise

- ▶ Create an algorithm to push a value into a sorted stack so that the stack remains sorted
 - Only use push and pop
 - You can use multiple stack to do it

Exercise

- Create an algorithm to check whether a word inputted from user is a palindrome or not
- Example
 - Input : 10011001
 - Output : palindrome

 - Input : Was it a car or a cat I saw
 - Output : palindrome

Algebraic Expression

- ▶ An algebraic expression is a legal combination of operands and the operators.
 - Operand is the quantity (unit of data) on which a mathematical operation is performed.
 - Operand may be a variable like x , y , z or a constant like 5, 4, 0, 9, 1 etc.
 - Operator is a symbol which signifies a mathematical or logical operation between the operands. Example of familiar operators include $+$, $-$, $*$, $/$, $^$
 - Considering these definitions of operands and operators now we can write an example of expression as $x + y * z$.

Infix, Postfix and Prefix Expressions

► **INFIX:**

- operands surround the operator,
- $x+y$, $6*3$ etc.

► **POSTFIX:**

- Reverse Polish Notation (RPN).
- operator comes after the operands,
- $xy+$, $xyz+*$ etc.

► **PREFIX:**

- Polish notation.
- operator comes before the operands,
- $+xy$, $*+xyz$ etc.

Properties

- ▶ Operator Priorities
 - $\text{priority}(\ast) = \text{priority}(/) > \text{priority}(+) = \text{priority}(-)$
- ▶ Tie Breaker
 - When an operand lies between two operators that have the same priority, the operand associates with the operator **on the left**.

Why use Prefix or Postfix

- ▶ **INFIX** notations are not as simple as they seem specially while evaluating them.
- ▶ To evaluate an infix expression we need to consider Operators' Priority and Associative property
 - **For example** expression $3+5*4$ evaluate to 32 i.e. $(3+5)*4$ or to 23 i.e. $3+(5*4)$.

Infix Expression Is Hard To Parse

- Need operator priorities, tie breaker, and delimiters.
- This makes computer evaluation more difficult than is necessary.
- **Postfix** and **prefix** expression forms do not rely on operator priorities, a tie breaker, or delimiters.
- So it is easier to evaluate expressions that are in these forms.

Examples of infix to prefix and post fix

Infix	PostFix	Prefix
$A+B$	$AB+$	$+AB$
$(A+B) * (C + D)$	$AB+CD+*$	$*+AB+CD$
$A-B/(C*D^E)$	$ABCDE^*/-$	$-A/B*C^DE$

No brackets necessary

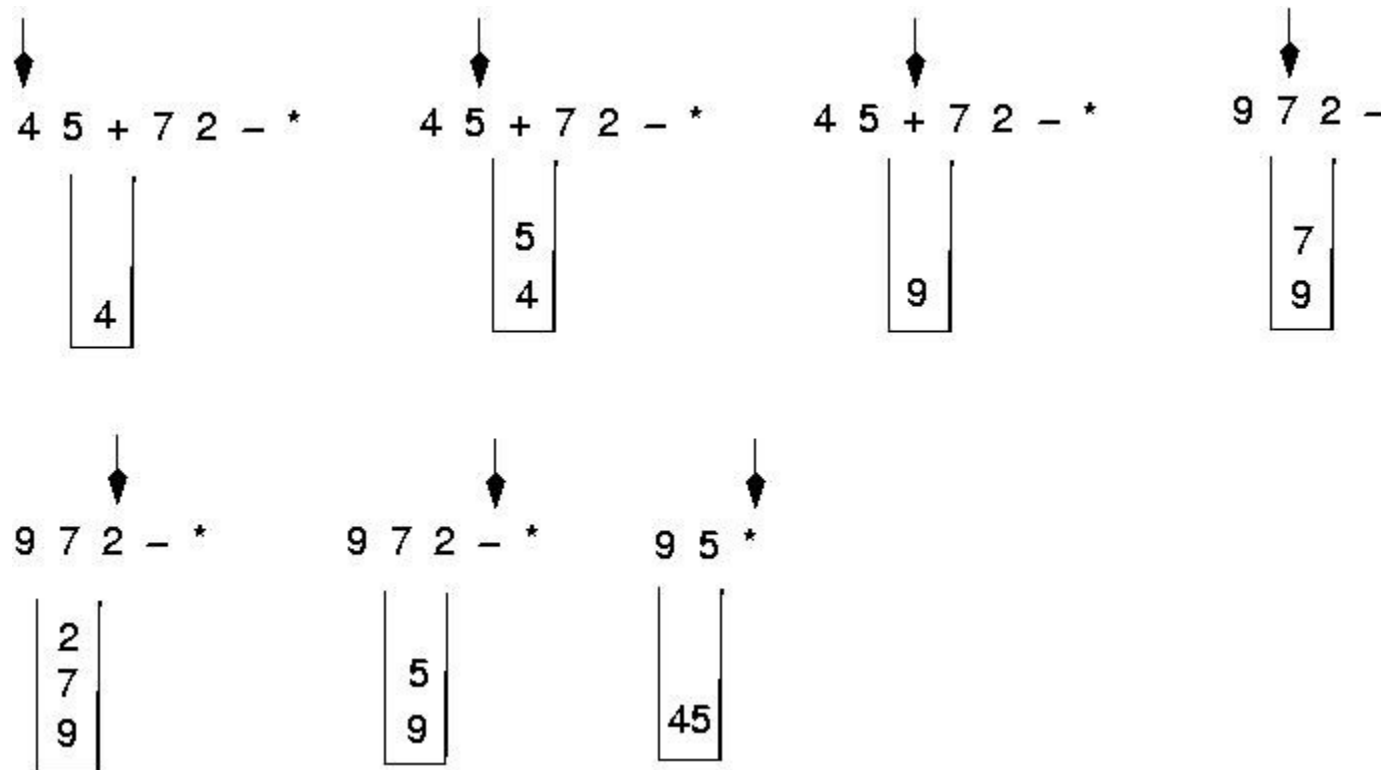
Example : Infix to Postfix

Expression		Stack	Output
2	(bottom stack)	Empty	2
*		*	2
3		*	23
/		/	23*
(/(23*
-		/(-	23*2
1		/(-	23*21
)		/	23*21-
+		+	23*21-/
5		+	23*21-/5
*		+*	23*21-/5
3		+*	23*21-/53
		Empty	23*21-/53*+

Suppose we want to convert $2*3/(2-1)+5*3$ into Postfix form,

Evaluating a postfix Expression

- Use a stack to evaluate an expression in postfix notation



Home Task

- Write an algorithm to convert an infix expression to postfix and infix to prefix
- Write an algorithm to evaluate a postfix and a prefix expression



Fakultas Informatika
School of Computing
Telkom University



THANK YOU