DATA STRUCTURES AND ALGORITHMS

BY

A.ROGHYA PARVEEN

ASSISTANT PROFESSOR IN DEPARTMENT OF COMPUTER SCIENCE & IT

JAMAL MOHAMED COLLEGE (AUTONOMOUS),TRICHY.

STACK

Stack

A stack is a data structure in which items can be inserted only from one end and get items back from the same end. There , the last item inserted into stack, is the the first item to be taken out from the stack. In short its also called Last in First out [LIFO].

Example of Stack (LIFO)

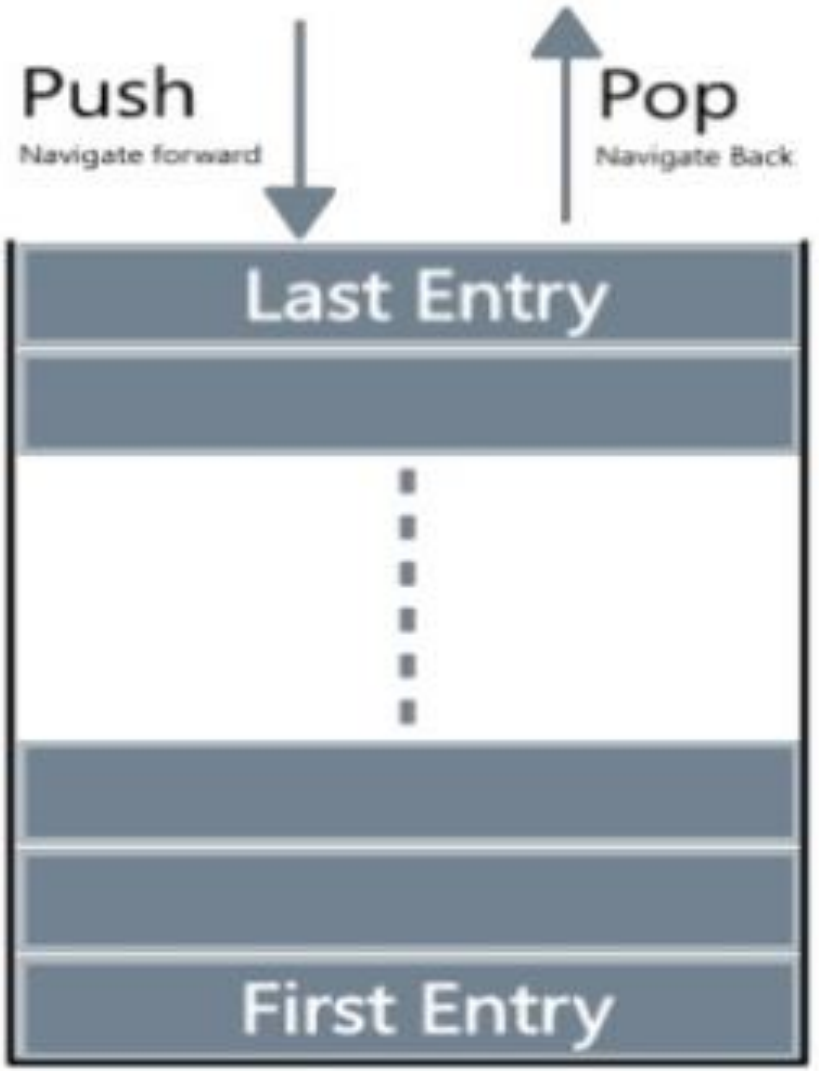
- A Stack of book on table.

- Token stack in Bank.

- Stack of trays and plates.

Operations that can be performed on STACK:

- PUSH.
- POP.



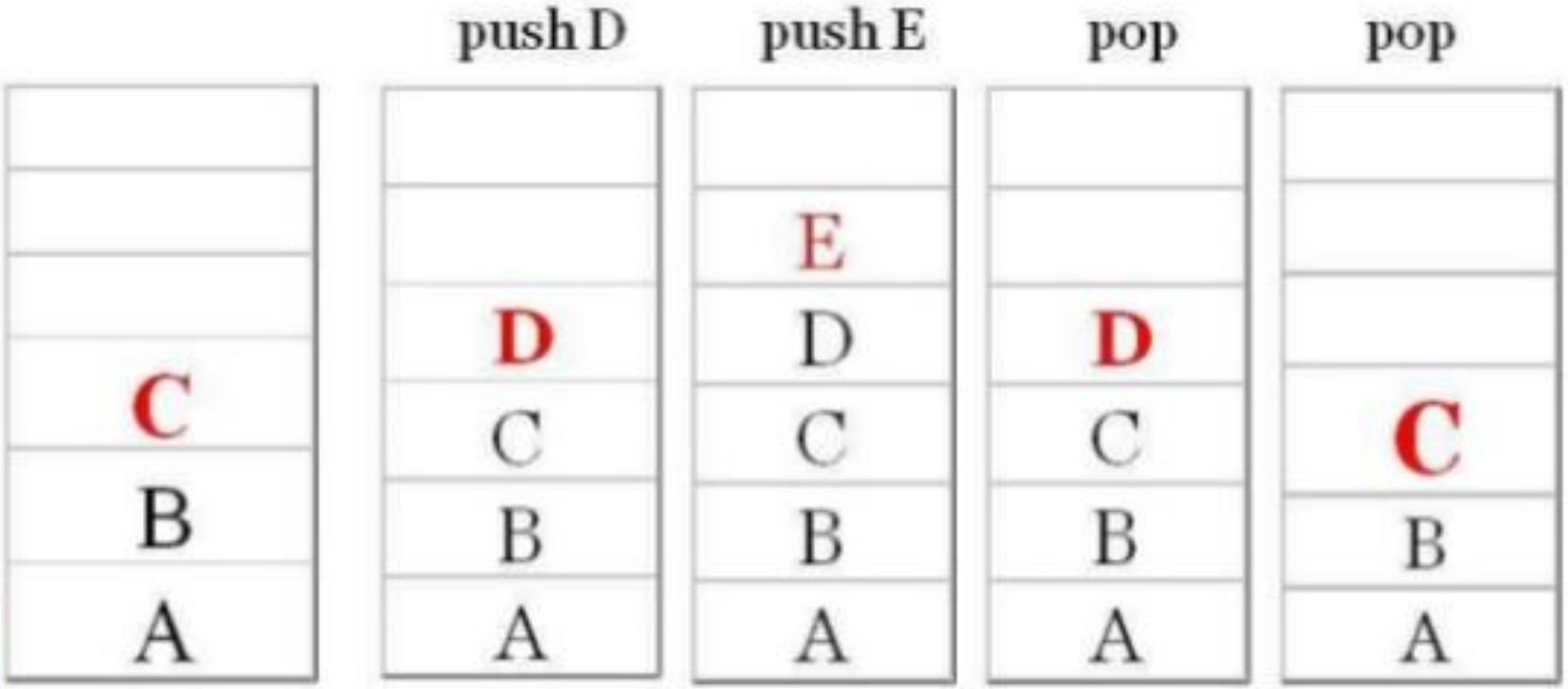
Stack Operations

- **Top:** Open end of the stack is called Top, From this end item can be inserted.
- **Push:** To insert an item from Top of stack is called push operation. The push operation change the position of Top in stack.
- **POP:** To put-off, get or remove some item from top of the stack is the pop operation, We can POP only only from top of the stack.
- **IsEmpty:** Stack considered empty when there is no item on Top. IsEmpty operation return true when no item in stack else false.
- **IsFull:** Stack considered full if no other element can be inserted on top of the stack. This condition normally occur when stack implement ed through array.

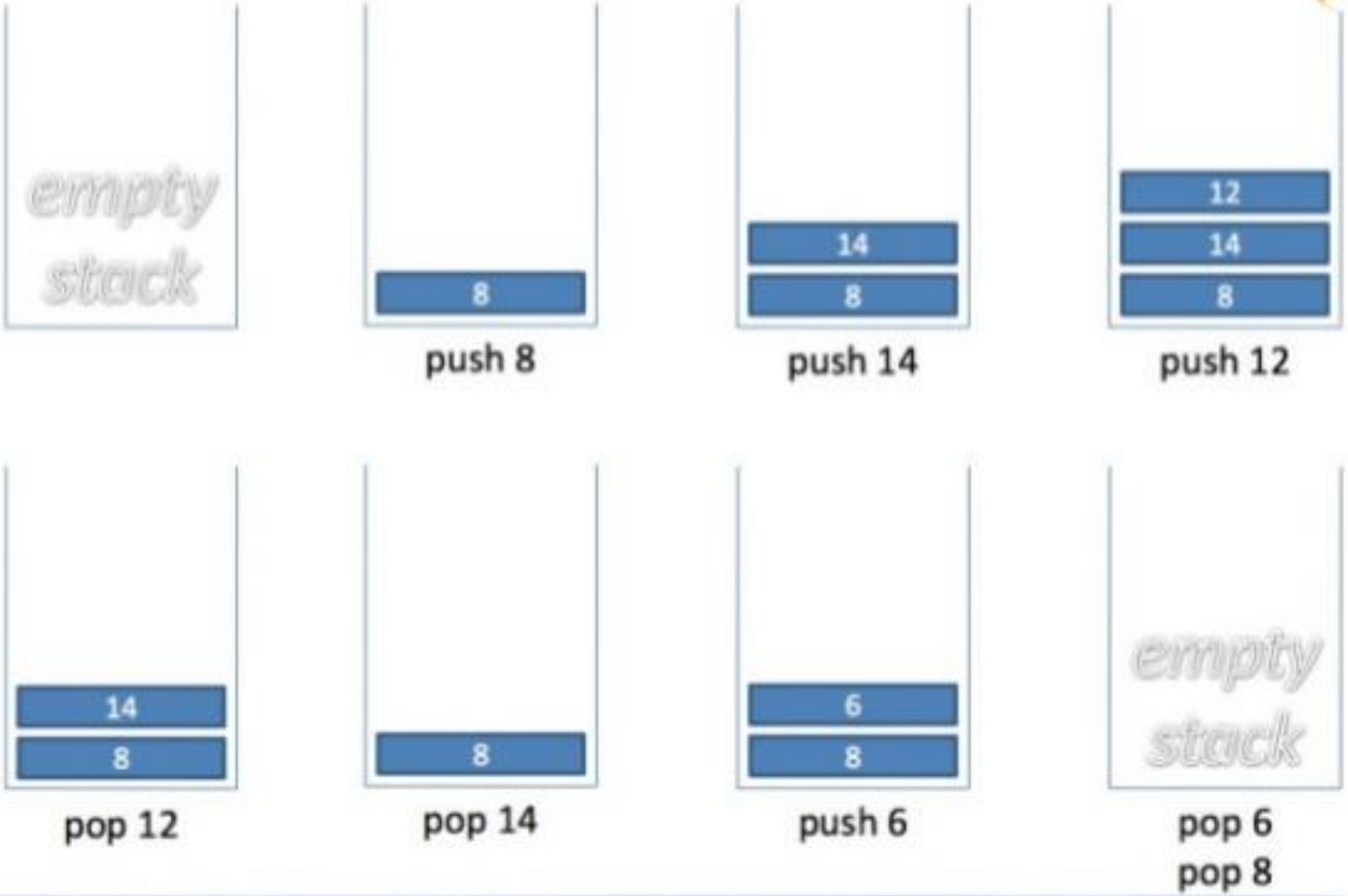
PUSH : It is used to insert items into the stack.

POP: It is used to delete items from stack.

TOP: It represents the current location of data in stack.



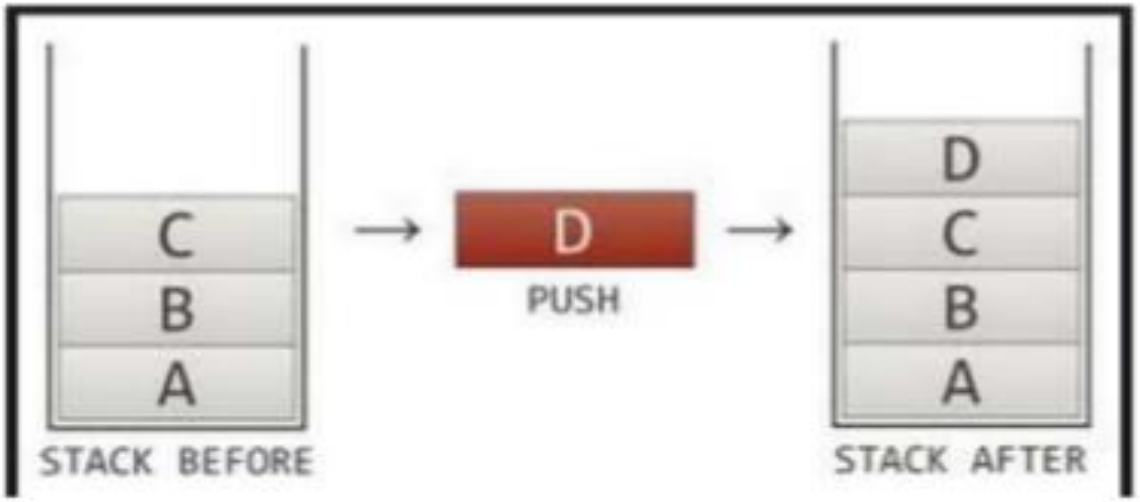
Stack Example



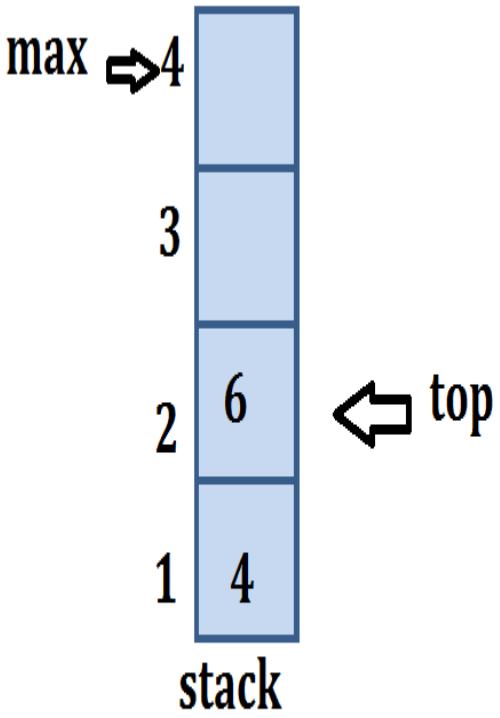
ALGORITHM OF INSERTION I
STACK: (PUSH)

Clip slide

1. Insertion(a,top,item,max)
2. If top=max then
   print 'STACK OVERFLOW'
   exit
   else
3. top=top+1
   end if
4. a[top]=item
5. Exit

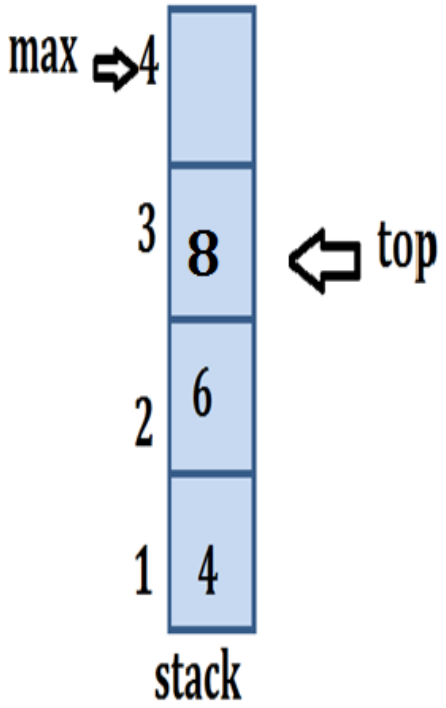


PUSH OPERATION



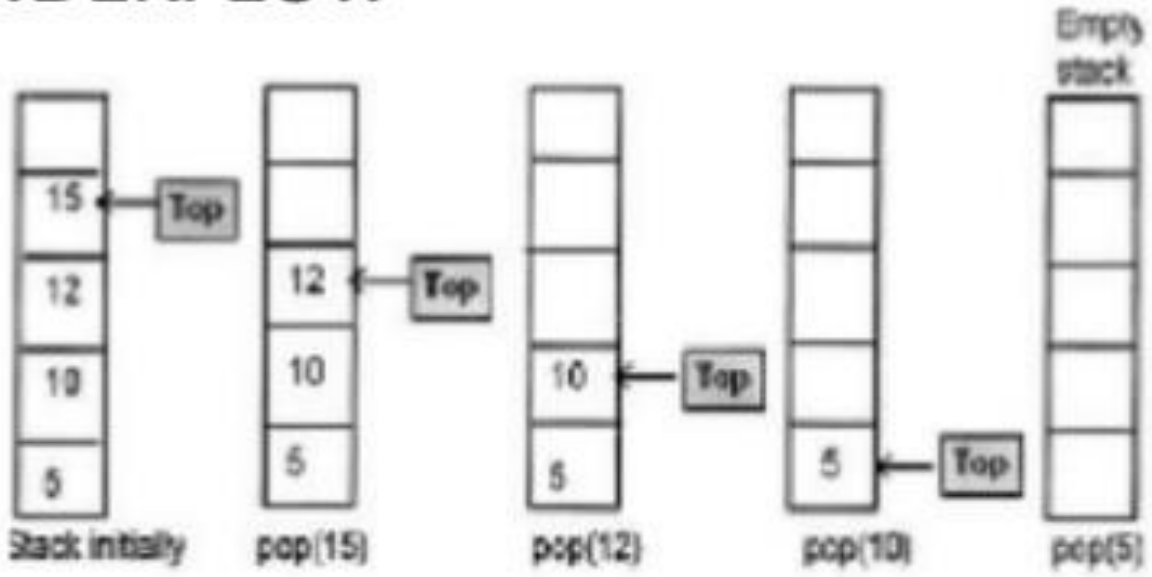
Let us consider stack as
PUSH(stack,2,8,4)
here top=2
max=4
item to be inserted=8
1.top=max
  2=4 false
2.top=top+1
  =2+1=3
 top=3
3.stack(top)=item
  stack(3)=8

thus item 8 is inserted in stack.

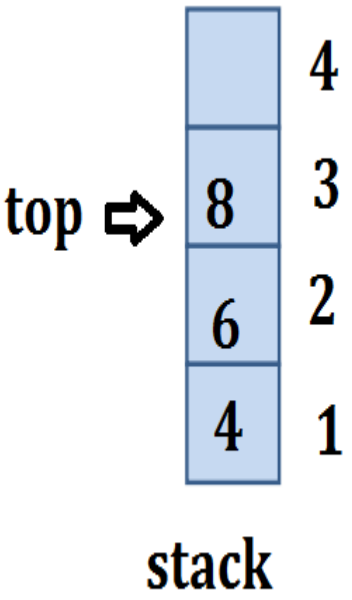


ALGORITHM OF DELETION IN STACK: (POP)

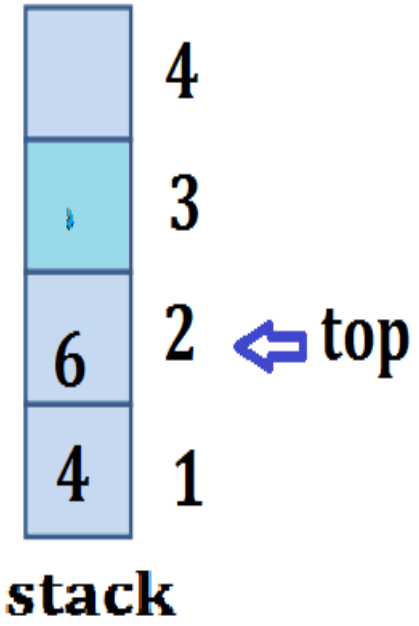
1. Deletion(a,top,item)
2. If top=0 then
   print 'STACK UNDERFLOW'
   exit
   else
3. item=a[top]
   end if
4. top=top-1
   Exit



Pop operation

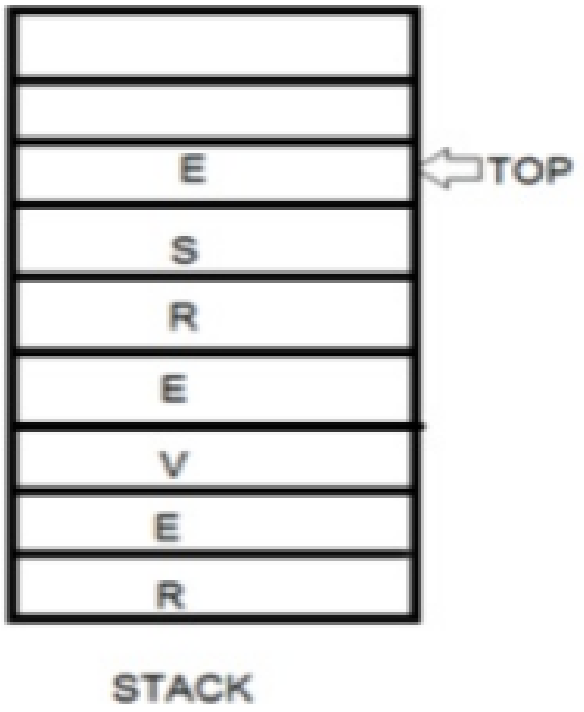


pop (Stack,3,8)
top=3 item=8
1.If top=0
3=0 false
2.Item=stack[top]
=stack[3]
=8
3.top=top-1
= 3-1=2



ALGORITHM OF DISPLAY IN STACK:

1.Display(top,i,a[i])
2.If top=0 then
 Print 'STACK EMPTY'
 Exit
 Else
3.For i=top to 0
  Print a[i]
  End for
4.exit



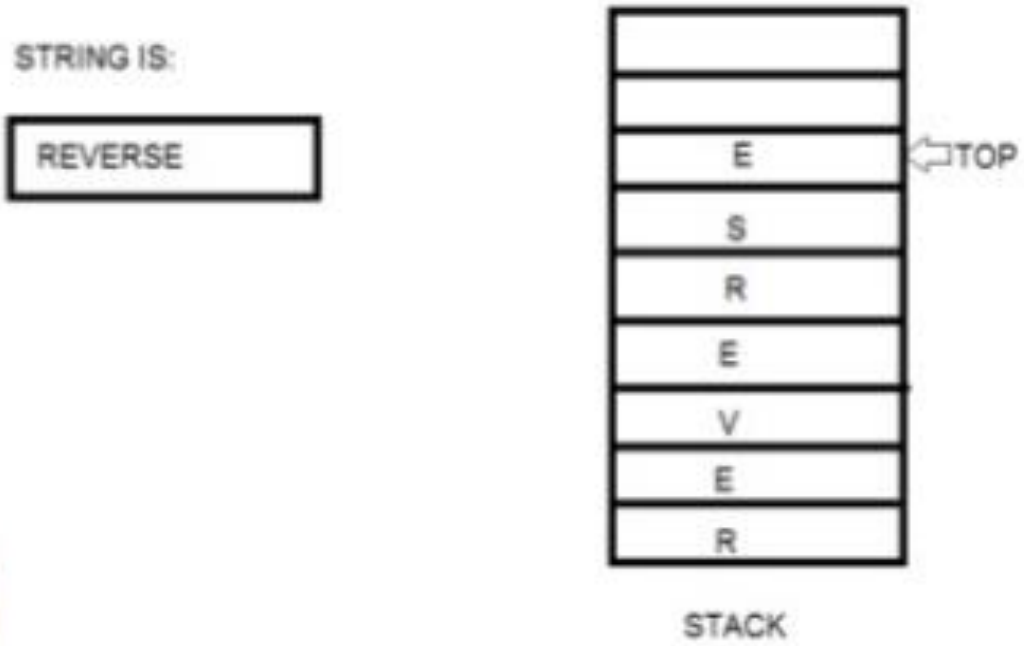
APPLICATIONS OF STACKS ARE:

I. Reversing Strings:

- A simple application of stack is reversing strings. To reverse a string , the characters of string are pushed onto the stack one by one as the string is read from left to right.
- Once all the characters of string are pushed onto stack, they are popped one by one. Since the character last pushed in comes out first, subsequent pop operation results in the reversal of the string.

For example:

To reverse the string 'REVERSE' the string is read from left to right and its characters are pushed . LIKE:



2.Evaluation of Expressions

Infix, Postfix and Prefix Expression

Infix Expression: An expression in which the operator is in between its two operands.

A+B

Prefix Expression: An expression in which operator precedes its two operands is called an prefix expression.

+AB

Postfix Expression: An expression in which operator follows its two operands is called a postfix expression.

AB+

Examples of Infix, Prefix, and Postfix expressions

Infix Expression	Prefix Expression	Postfix Expression
A + B	+ A B	A B +
A + B * C	+ A * B C	A B C * +
A + B * C + D	+ + A * B C D	A B C * + D +
(A + B) * (C + D)	* + A B + C D	A B + C D + *
A * B + C * D	+ * A B * C D	A B * C D * +
A + B + C + D	+ + + A B C D	A B + C + D +

Evaluation of Postfix Expression

Algorithm

Step 1: maintain a stack and scan the postfix expression from left to right

Step 2: If the element is a number, push it into the stack

Step 3: If the element is a operator ,

a)pop twice and get A and B respectively.

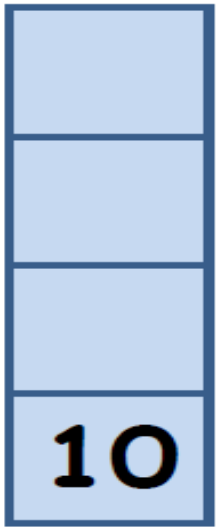
b) Calculate B operator A and push it back to the stack.

Step 4: When the expression is ended, the number in the stack is the final answer

Evaluation of Postfix Expression

Ex: 10 2 8 * + 3 –

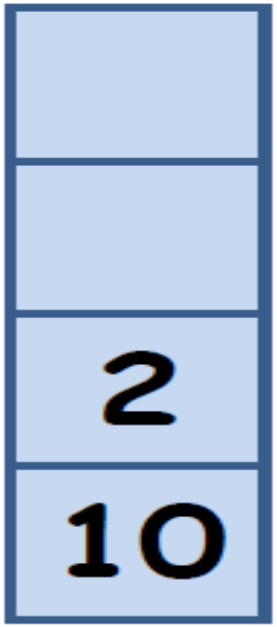
First, push(10) into the stack



Evaluation of Postfix Expression

Ex: 10 2 8 * + 3 -

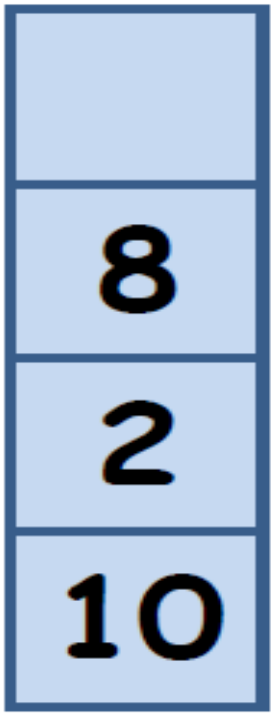
Then, push(2) into the stack



Evaluation of Postfix Expression

Ex: 10 2 8 * + 3 -

Push(8) into the stack



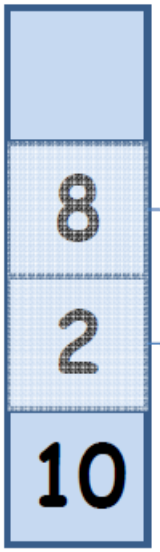
Evaluation of Postfix Expression

Ex: 10 2 8 * + 3 -

- **Now we see an operator *,that means we can get an new number by calculation**

Evaluation of Postfix Expression

Ex: 10 2 8 * + 3 -



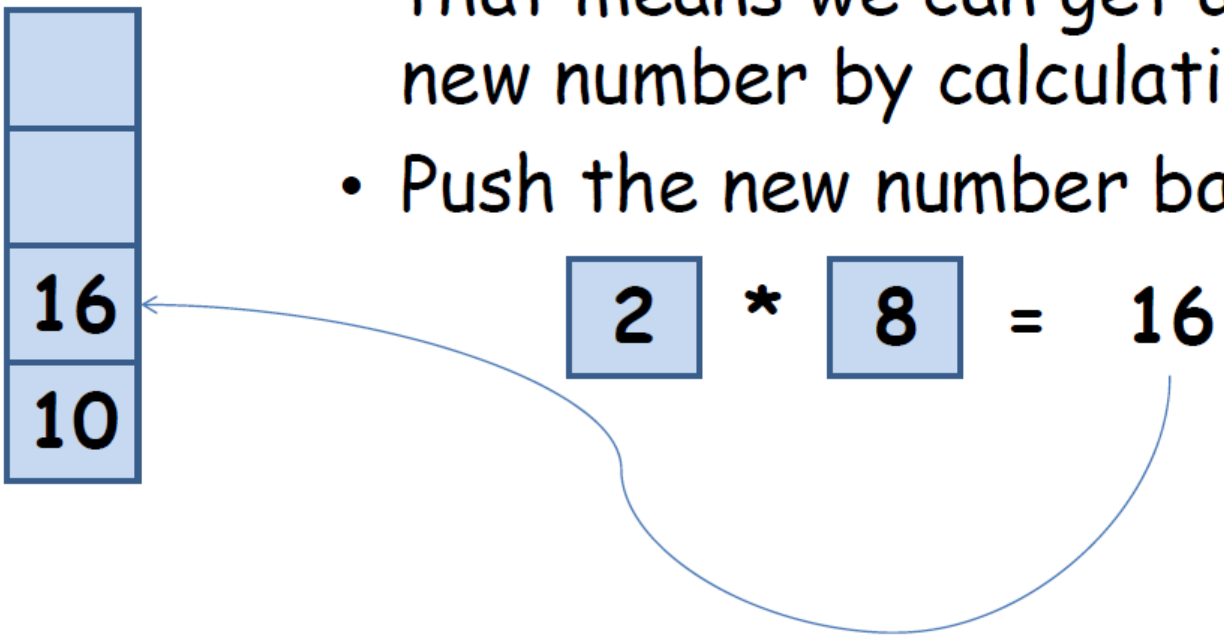
- Now we see an operator *, that means we can get an new number by calculation
- Pop the first two numbers

2 * 8 = 16

Evaluation of Postfix Expression

Ex: 10 2 8 * + 3 -

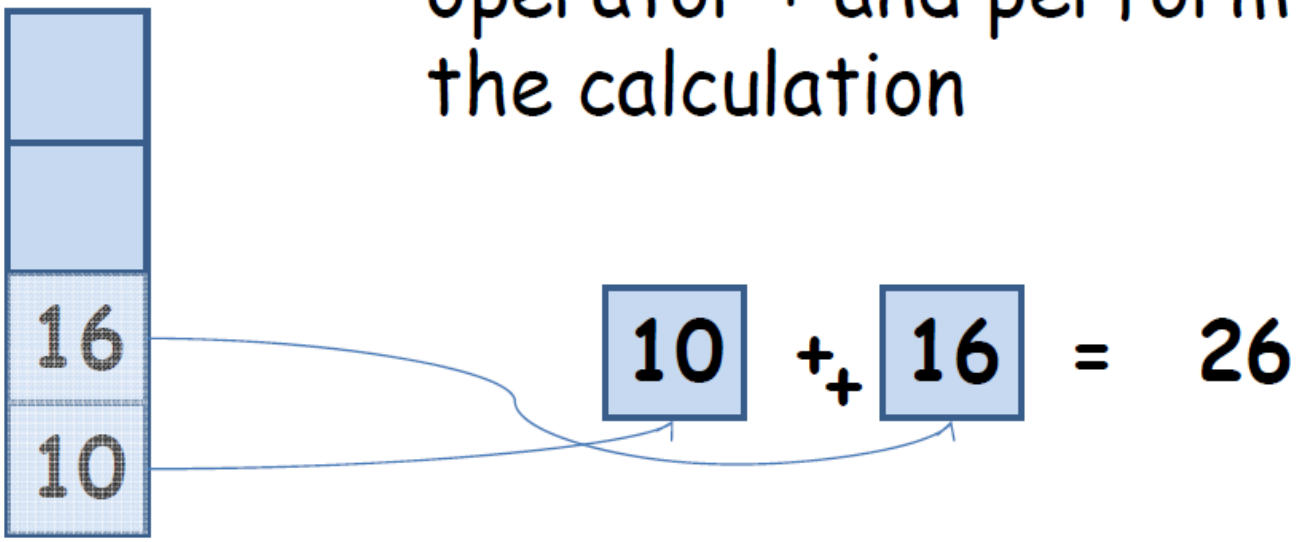
- Now we see an operator *, that means we can get an new number by calculation
- Push the new number back



Evaluation of Postfix Expression

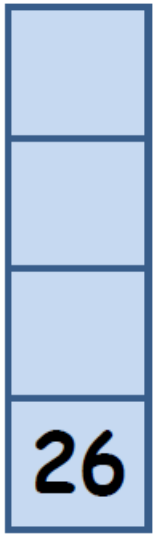
Ex: 10 2 8 * + 3 -

- Then we see the next operator + and perform the calculation



Evaluation of Postfix Expression

Ex: 10 2 8 * + 3 -



- Then we see the next operator + and perform the calculation
- Push the new number back

10 + **16** = 26

Evaluation of Postfix Expression

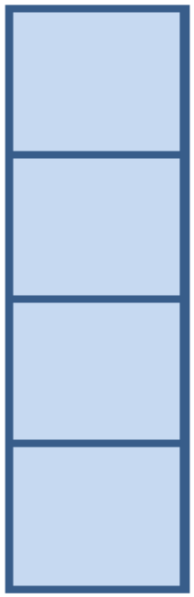
Ex: 10 2 8 * + 3 -

- **We see the next number 3**
- **Push (3) into the stack**

Evaluation of Postfix Expression

Ex: 10 2 8 * + 3 -

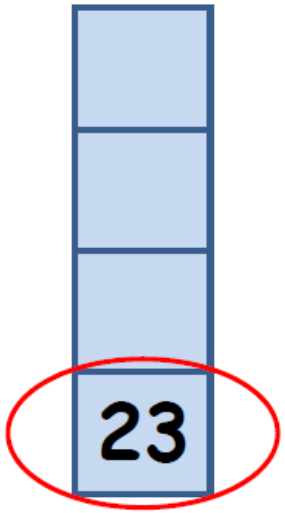
- **The last operation**



Evaluation of Postfix Expression

Ex: 10 2 8 * + 3 -

- The last operation



26 - 3 = 23

answer!

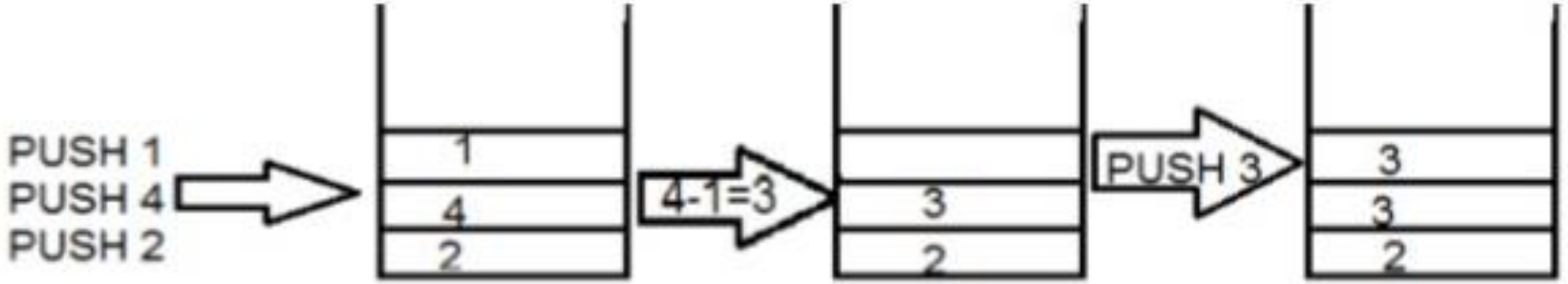
Conversion of INFIX to POSTFIX conversion

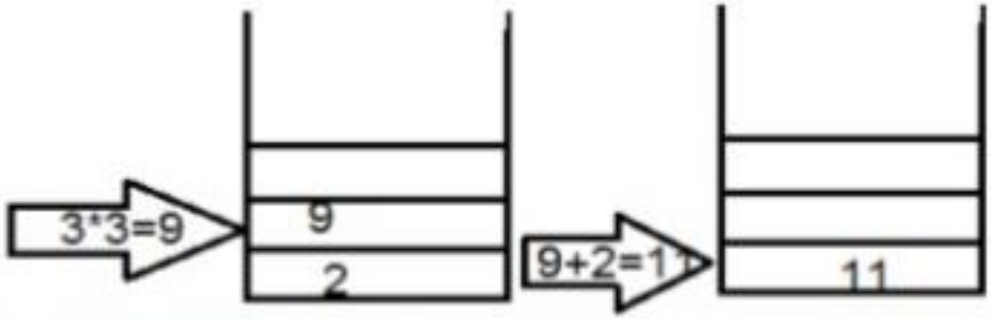
Example: 2+(4-1)*3 step1

2+41-*3 step2

2+41-3* step3

241-3*+ step4





CONVERSION OF INFIX INTO POSTFIX

2+(4-1)*3 into 241-3*+

CURRENT SYMBOL	ACTION PERFORMED	STACK STATUS	POSTFIX EXPRESSION
(	PUSH C	C	2
2			2
+	PUSH +	(+	2
(	PUSH (	(+(	24
4			24
-	PUSH -	(+(-	241
1	POP		241-
)		(+	241-
*	PUSH *	(+*	241-
3			241-3
	POP *		241-3*
	POP +		241-3*+
)			

REFERENCES

Books

1. Fundamentals of Data Structure – Ellis Horowiz, SartajSahni, Galgotia

Publications, 2008.

2. Data Structures – Seymour Lipschutz, Tata Mcgraw Hill, Schaum's Outline

Series, 2014.

Websites

1.https://www.learnpick.in/prime/documents/ppts/details/17111/Stack