



CS & IT ENGINEERING

Data Structures

Stack and Queues

Lecture No.- 04

By- Pankaj Sharma Sir



Recap of Previous Lecture



Topic

Stack and Queues Part - 03



Topics to be Covered



Topic

Stack and Queues Part - 04

Questions



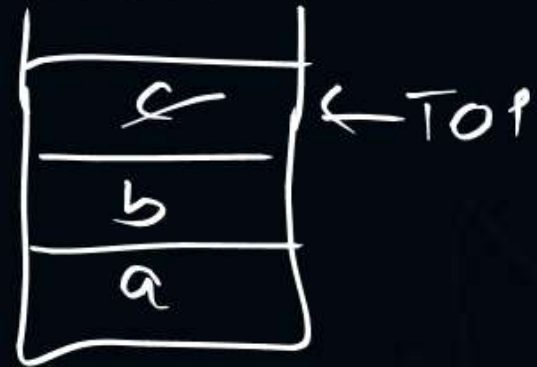


Topic : Stack and Queues

#Q. A program attempts to generate as many permutations as possible of the string "abcd" by pushing the character a, b, c, d in the same order onto a stack but it may pop off the top character at any time. Which one of the following strings CANNOT be generated using this program?

- A. abcd ✓
- B. dcba ✓
- C. cbad ✓
- D. cabd ✗

push a
push b
push c
Pop





Topic : Stack and Queues

#Q. The postfix expression corresponding to the infix expression
 $a + b * c - d \wedge e \wedge f$ is_____.

$$a + b * c - d \wedge [e f \wedge]$$

$$a + \underbrace{b * c} - \underbrace{[d e f \wedge \wedge]}$$

$$\underbrace{a + [b * c]} - [d e f \wedge \wedge]$$

$$[a b c * +] - [d e f \wedge \wedge]$$

$$a b c * + d e f \wedge \wedge -$$



Topic : Stack and Queues

#Q. The following postfix expression with single digit operand is evaluated using a stack:

$$8\ 2\ 3\ ^\wedge / 2\ 3\ * + 5\ 1\ * -$$

Note that $^\wedge$ is the exponentiation operator. The top two elements of the stack after the first $*$ is evaluated are:

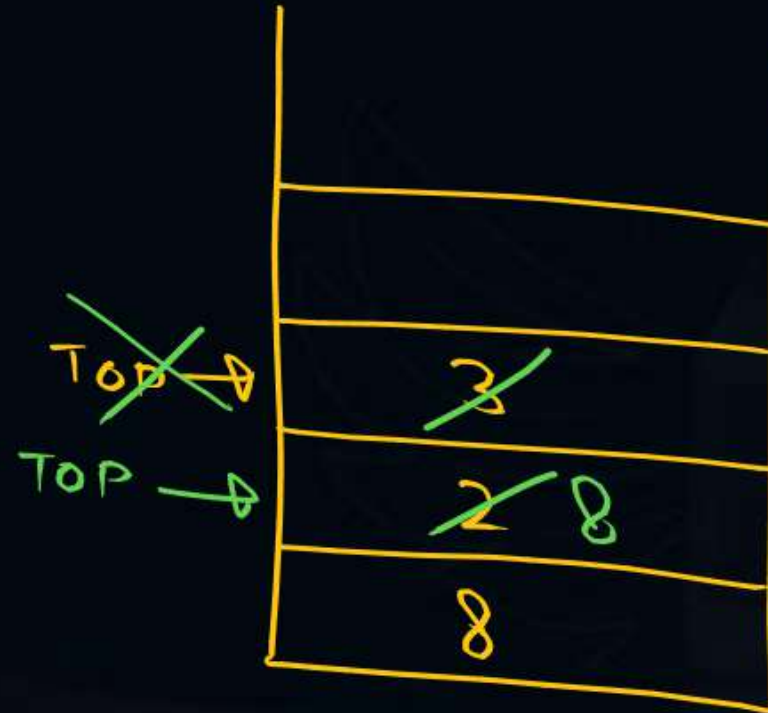
A. 6, 1

B. 5, 7

C. 3, 2

D. 1, 5

$^\wedge$
(i) Pop 3
(ii) Pop 2
3, 2
 2^3
 \Rightarrow Push 8





Topic : Stack and Queues

#Q. The following postfix expression with single digit operand is evaluated using a stack:

$$8\ 2\ 3\ ^\wedge / 2\ 3\ * + 5\ 1\ * -$$

Note that $^$ is the exponentiation operator. The top two elements of the stack after the first $*$ is evaluated are:

A. 6, 1

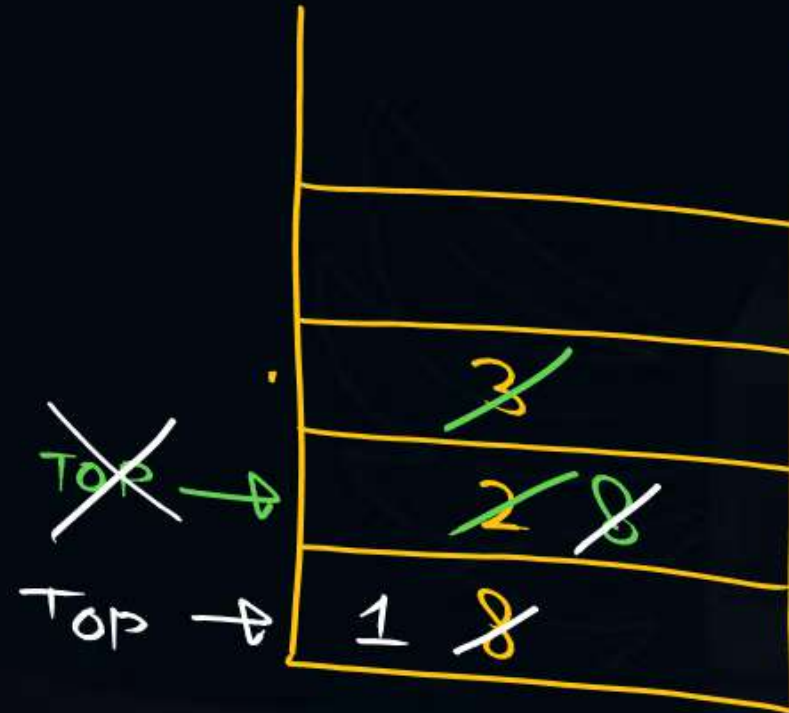
B. 5, 7

C. 3, 2

D. 1, 5

/

(i) Pop 8
(ii) Pop 8
8, 8
8/8
→ Push 1





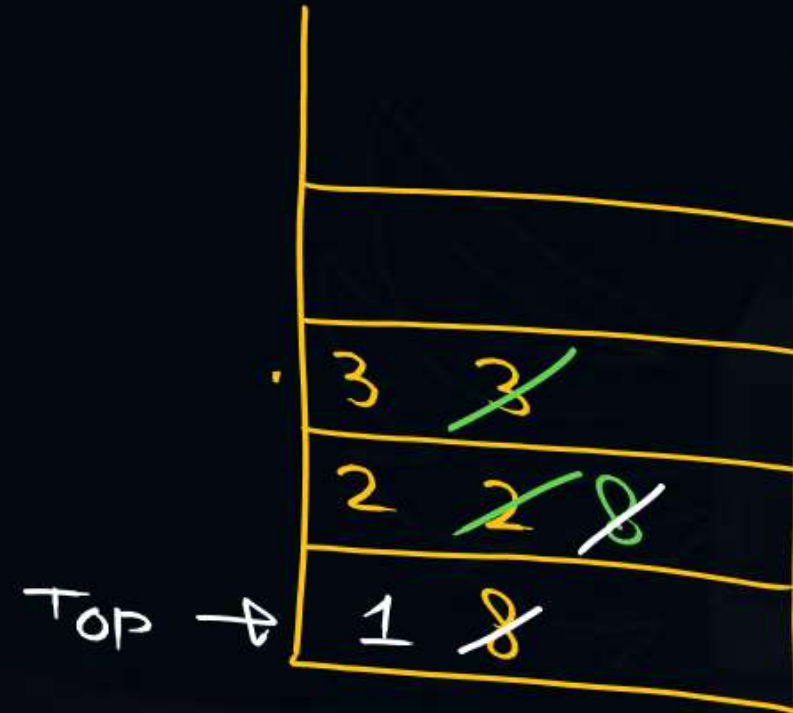
Topic : Stack and Queues

#Q. The following postfix expression with single digit operand is evaluated using a stack:

$$8\ 2\ 3\ ^\ / \ 2\ 3\ ^\ +\ 5\ 1\ ^\ -$$

Note that ^ is the exponentiation operator. The top two elements of the stack after the first * is evaluated are:

- A. 6, 1
- B. 5, 7
- C. 3, 2
- D. 1, 5





Topic : Stack and Queues

#Q. The following postfix expression with single digit operand is evaluated using a stack:

$$8\ 2\ 3\ ^\wedge / 2\ 3\ ^\wedge + 5\ 1\ ^\wedge -$$

Note that $^$ is the exponentiation operator. The top two elements of the stack after the first $^$ is evaluated are:

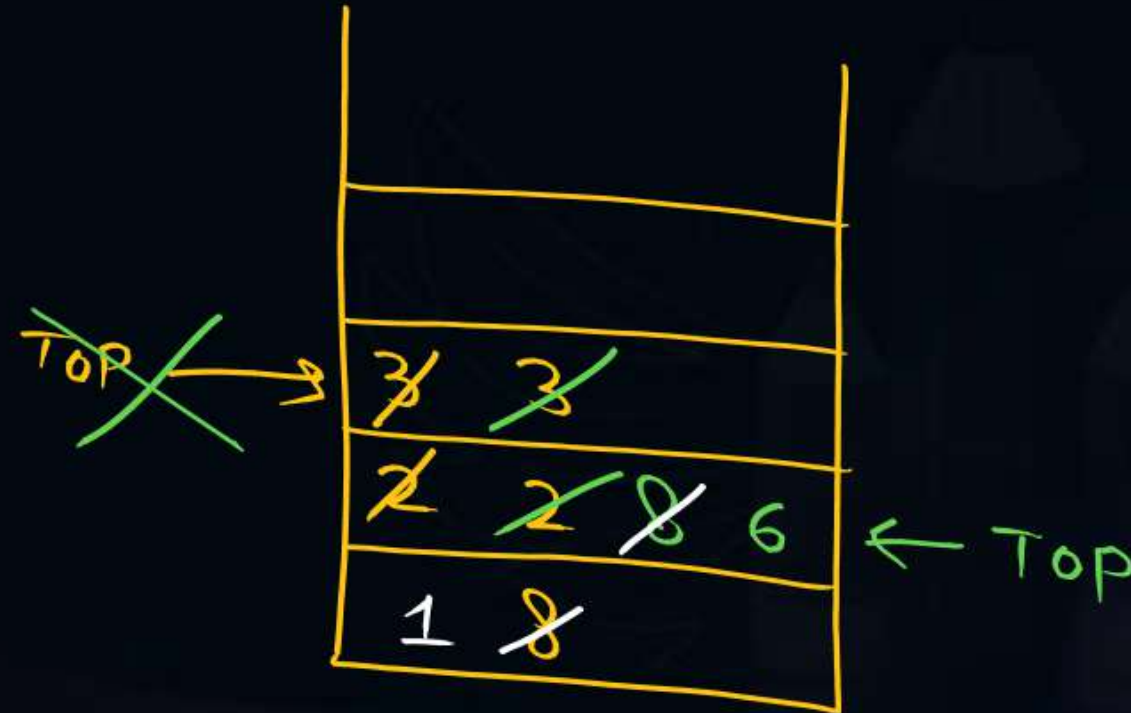
A. 6, 1

B. 5, 7

C. 3, 2

D. 1, 5

$^$
(i) Pop 3
(ii) Pop 2
3 2
2 x 3
Push 6





Topic : Stack and Queues



#Q. Consider the following C program:

```
#include <stdio.h>
#define EOF - 1
void push (int);
/*push the argument on the stack*/
int pop(void); /* pop the top of the stack*/
void flagError();
```

postfix eval.

```
int main()
{
    int c, m, n, r;
    while((c = getchar()) != EOF)
    {
        if(isdigit(c) push(c);
        else if ((c == '+' || (c == '*'))
        {
            m = pop();
            n = pop();
            r = (c == '+')? n + m : n*m;
            push(r);
        }else if (c != '"')
            flagError();
        }
        printf("%c", pop());
    }
}
```


What is the output of the program for the following?

5 2 * 3 3 2 + * +



* +

digit

5 2 x 3 3 2 + * +

10 3 3 2 + x +

10 3 5 x +

10 15 +

25

2
5

8



Topic : Stack and Queues

5 min ✓✓



#Q. Let S be a stack of size $n \geq 1$. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that PUSH and POP operations take X secs each and Y seconds elapse between that end of one such stack operation and the start of the next operation. For $m \geq 1$, define the stack life-time of m as the time elapsed from the end of PUSH(m) to the start of POP operation that removes m from S . the average stack-life of an element is-

- A. $n(X + Y)$ B. $n(X + Y) - X$
C. $3Y + 2X$ D. $Y + 2X$

Push/Pop $\rightarrow X$ sec

Push, Push
Push, Pop
Pop, Push
Pop, Pop

} Y sec

$$\text{slack_life}(m) = \text{Pop_start_time}(m) \\ - \text{Push_end_time}(m)$$



Topic : Stack and Queues

#Q. Let S be a stack of size $n \geq 1$. Starting with the empty stack, suppose we push the first n natural numbers in sequence, and then perform n pop operations. Assume that PUSH and POP operations take X secs each and Y seconds elapse between that end of one such stack operation and the start of the next operation. For $m \geq 1$, define the stack life-time of m as the time elapsed from the end of PUSH(m) to the start of POP operation that removes m from S . the average stack-life of an element is-


~~A.~~ $n(X + Y)$

☒ B. $n(X + Y) - X$

~~C.~~ $3Y + 2X$

~~D.~~ $Y + 2X$

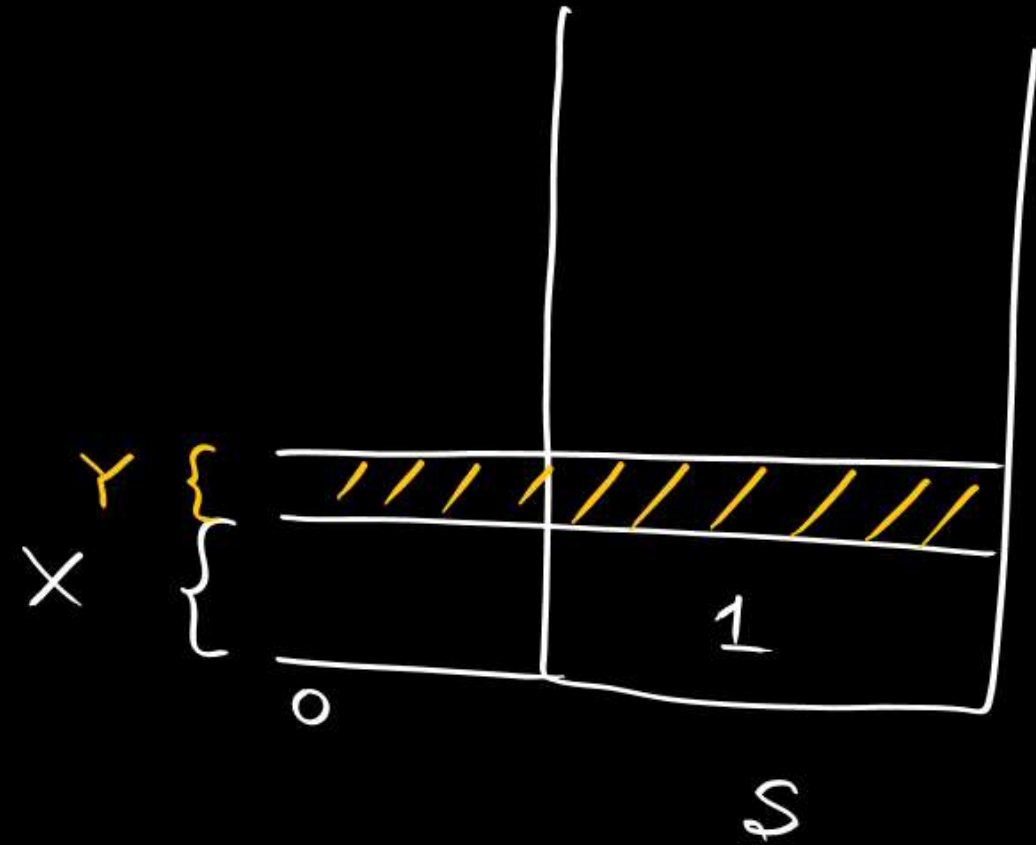
Topic : Easy \rightarrow Twist

Stack/Queue \rightarrow 
long.

$n=1$

	start	End
Push(1)	0	X

Pop(1)	$X+Y$	$2X+Y$
--------	-------	--------



Stack life of 1

$$\Rightarrow \boxed{(X+Y) - X}$$

Pop \rightarrow will start $(X+Y)$

Pop \rightarrow End $X+Y+X$



Topic : Stack and Queues



$$\left(2 - (5 + 1) - 7 \right) \times 3$$

#Q. The attributes of three arithmetic operators in some programming language are given below.

Operator	Precedence	Associativity	Arity
+	High	Left	Binary
-	Medium	Right	Binary
*	Low	Left	Binary

$$\begin{aligned} & 6 - 7 \\ & (2 - (-1)) \times 3 \\ & 3 \times 3 \\ & = 9 \end{aligned}$$

The value of the expression $2 - 5 + 1 - 7 * 3$ in this language is 9.

Q) A func. f defined on stacks of integers satisfies:

$$f(\emptyset) = 0 \quad \text{and} \quad f(\text{Push}(S, i)) = \max(f(S), 0) + i$$

for all stacks S &
integer i .

If a stack S contains the integers $2, -3, 2, -1, 2$ in order from bottom to top, what is $f(S)$?

A) 6

D) 2

B) 4

C) 3

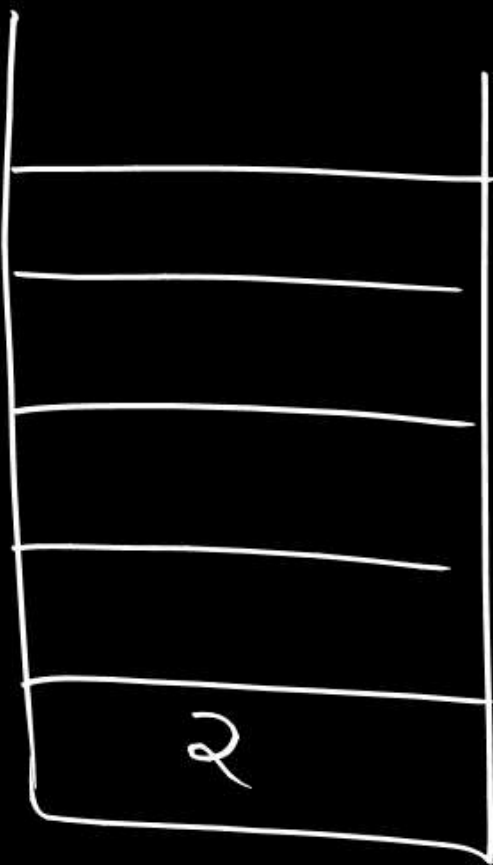
Q) A func. f defined on stacks of integers satisfies:

$$f(\emptyset) = 0 \quad \text{and} \quad f(\text{Push}(s, i)) = \max(f(s), 0) + i$$

$$\boxed{f(s) = 0}$$

$f(\text{stack is empty}) = 0$

$$(i) \quad f(\text{Push}(s, 2)) = \max(0, 0) + 2 \\ = 2$$



for all stacks s &
integer i .

2, -3, 2, -1, 2

Q) A func. f defined on stacks of integers satisfies:

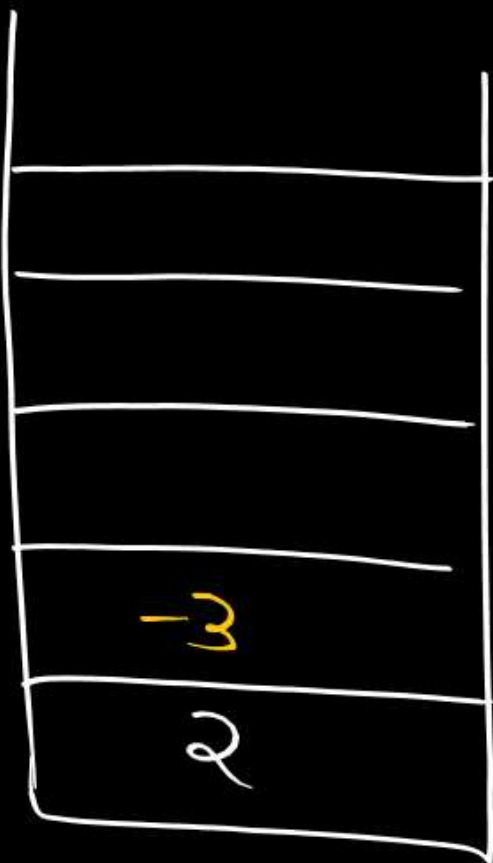
$$f(\emptyset) = 0 \quad \text{and} \quad f(\text{Push}(s, i)) = \max(f(s), 0) + i$$

$$\boxed{f(s) = 0}$$

$f(\text{stack is empty}) = 0$

$$(i) \quad f(\text{Push}(s, 2)) = \max(0, 0) + 2 \\ = 2$$

$$(ii) \quad f(\text{Push}(s, -3)) = \max(2, 0) + (-3) \\ = 2 + (-3) \\ = -1$$



for all stacks s &
integer i .

✓ ✓
2, -3, 2, -1, 2

Q) A func. f defined on stacks of integers satisfies:

$$f(\emptyset) = 0 \quad \text{and} \quad f(\text{Push}(s, i)) = \max(f(s), 0) + i$$

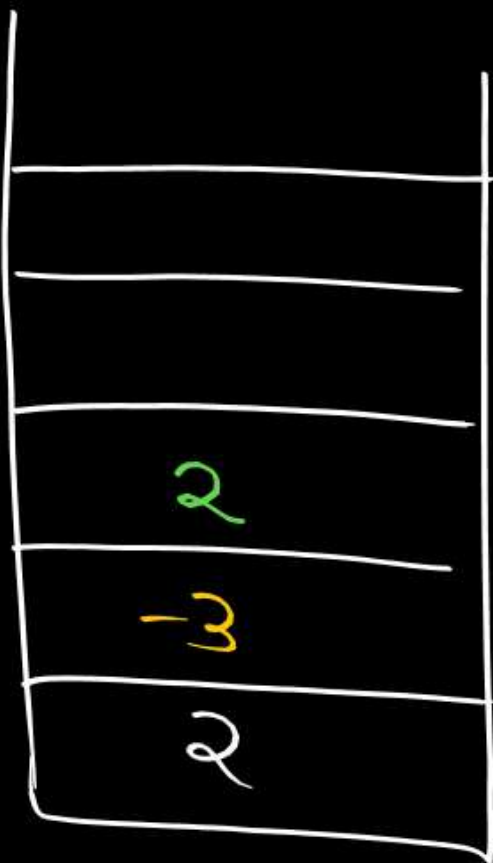
$$\boxed{f(s) = 0}$$

$f(\text{stack is empty}) = 0$

$$(i) \quad f(\text{Push}(s, 2)) = \max(0, 0) + 2 \\ = 2$$

$$(ii) \quad f(\text{Push}(s, -3)) = \max(2, 0) + (-3) \\ = 2 + (-3) \\ = -1$$

$$(iii) \quad f(\text{Push}(s, 2)) = \max(-1, 0) + 2 \\ = 0 + 2 \\ = 2$$



for all stacks s &
integer i .

✓ ✓ ✓
2, -3, 2, -1, 2

$$(iv) f(\text{push}(s, -1)) = \max(2, 0) + (-1) \\ = 2 + (-1) = 1$$

$$\boxed{f(s) = 0}$$

$$f(\text{stack is empty}) = 0$$

$$(i) f(\text{push}(s, 2)) = \max(0, 0) + 2 \\ = 2$$

$$ii) f(\text{push}(s, -3)) = \max(2, 0) + (-3) \\ = 2 + (-3) \\ = -1$$

$$(iii) f(\text{push}(s, 2)) = \max(-1, 0) + 2 \\ = 0 + 2 \\ = 2$$

-1
2
-3
2

✓ ✓ ✓ ✓
2, -3, 2, -1, 2

$$(iv) f(\text{Push}(s, -1)) = \max(2, 0) + (-1) \\ = 2 + (-1) = 1$$

$$(v) f(\text{Push}(s, 2)) = \max(1, 0) + 2 \\ = 1 + 2 \\ = \textcircled{3}$$

✓ ✓ ✓ ✓ ✓
2, -3, 2, -1, 2

2
-1
2
-3
2

$$\boxed{f(s) = 0}$$

$$f(\text{stack is empty}) = 0$$

$$(i) f(\text{Push}(s, 2)) = \max(0, 0) + 2 \\ = 2$$

$$ii) f(\text{Push}(s, -3)) = \max(2, 0) + (-3) \\ = 2 + (-3) \\ = -1$$

$$(iii) f(\text{Push}(s, 2)) = \max(-1, 0) + 2 \\ = 0 + 2 \\ = 2$$

NAT
2015

```
int *A, stkTOP;
```

```
int stkFun( int opcode, int val)
```

```
{
```

```
static int size = 0, stkTOP = 0;
```

```
switch( opcode) {
```

```
case -1 :
```

```
size = val;  
break;
```

set

case (-1) : stack size

0 :

Push

default :

if non-empty stack

(pop)

stack
imp with
3 choices

```
case 0% if( stkTOP < size)
```

```
A[stkTOP++] = val;
```

```
break;
```

```
default : if( stkTOP)
```

```
return A[--stkTOP];
```

```
}
```

```
return -1;
```

```
}
```

```
int main() { int B[20]; A = B;  
stkTOP = -1;
```

```
stkFun(-1, 10); → set size = 10
```

```
stkFun(0, 5); →
```

```
stkFun(0, 10); →
```

```
printf("%d", stkFun(1, 0) + stkFun(1, 0));
```

10 + 5

15



THANK - YOU