

Our Solution(s)	Run Code	Your Solutions	Run Code
-----------------	----------	----------------	----------

Solution 1

```
1 // Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 package main
4
5 type MinMaxStack struct {
6     stack []int
7     minMaxStack []entry
8 }
9
10 type entry struct {
11     min int
12     max int
13 }
14
15 func NewMinMaxStack() *MinMaxStack {
16     return &MinMaxStack{}
17 }
18
19 // O(1) time | O(1) space
20 func (stack *MinMaxStack) Peek() int {
21     return stack.stack[len(stack.stack)-1]
22 }
23
24 // O(1) time | O(1) space
25 func (stack *MinMaxStack) Pop() int {
26     stack.minMaxStack = stack.minMaxStack[:len(stack.minMaxStack)-1]
27     out := stack.stack[len(stack.stack)-1]
28     stack.stack = stack.stack[:len(stack.stack)-1]
29     return out
30 }
31
32 // O(1) time | O(1) space
33 func (stack *MinMaxStack) Push(number int) {
```

Solution 1

Solution 2

Solution 3

```
1 package main
2
3 type MinMaxStack struct {
4     // Write your code here.
5 }
6
7 func NewMinMaxStack() *MinMaxStack {
8     // Write your code here.
9     return nil
10 }
11
12 func (stack *MinMaxStack) Peek() int {
13     // Write your code here.
14     return -1
15 }
16
17 func (stack *MinMaxStack) Pop() int {
18     // Write your code here.
19     return -1
20 }
21
22 func (stack *MinMaxStack) Push(number int) {
23     // Write your code here.
24 }
25
26 func (stack *MinMaxStack) GetMin() int {
27     // Write your code here.
28     return -1
29 }
30
31 func (stack *MinMaxStack) GetMax() int {
32     // Write your code here.
33     return -1
```

```
11 def isPrime(n):
12     if n < 2:
13         return False
14     if n == 2:
15         return True
16     if n % 2 == 0:
17         return False
18     for i in range(3, int(n**0.5) + 1, 2):
19         if n % i == 0:
20             return False
21     return True
22
23 def sieve(n):
24     isPrime = [True] * (n + 1)
25     isPrime[0] = False
26     isPrime[1] = False
27     for i in range(2, int(n**0.5) + 1):
28         if isPrime[i]:
29             for j in range(i*i, n + 1, i):
30                 isPrime[j] = False
31     return isPrime
32
33 def main():
34     n = int(input())
35     isPrime = sieve(n)
36     for i in range(2, n + 1):
37         if isPrime[i]:
38             print(i, end=" ")
39     print()
```

Run or submit code when you're ready.