

Our Solution(s)

Run Code

Solution 1

```
1 # Copyright © 2020 AlgoExpert, LLC. All rights reserved.
2
3 class MinMaxStack:
4     def __init__(self):
5         self.minMaxStack = []
6         self.stack = []
7
8     # O(1) time | O(1) space
9     def peek(self):
10         return self.stack[len(self.stack) - 1]
11
12     # O(1) time | O(1) space
13     def pop(self):
14         self.minMaxStack.pop()
15         return self.stack.pop()
16
17     # O(1) time | O(1) space
18     def push(self, number):
19         newMinMax = {"min": number, "max": number}
20         if len(self.minMaxStack):
21             lastMinMax = self.minMaxStack[len(self.minMaxStack) - 1]
22             newMinMax["min"] = min(lastMinMax["min"], number)
23             newMinMax["max"] = max(lastMinMax["max"], number)
24         self.minMaxStack.append(newMinMax)
25         self.stack.append(number)
26
27     # O(1) time | O(1) space
28     def getMin(self):
29         return self.minMaxStack[len(self.minMaxStack) - 1]["min"]
30
31     # O(1) time | O(1) space
32     def getMax(self):
33         return self.minMaxStack[len(self.minMaxStack) - 1]["max"]
```

Our Tests

Your Solutions

Run Code

Solution 1   Solution 2   Solution 3

```
1 # Feel free to add new properties and methods to the class.
2 class MinMaxStack:
3     def peek(self):
4         # Write your code here.
5         pass
6
7     def pop(self):
8         # Write your code here.
9         pass
10
11     def push(self, number):
12         # Write your code here.
13         pass
14
15     def getMin(self):
16         # Write your code here.
17         pass
18
19     def getMax(self):
20         # Write your code here.
21         pass
22
```

Custom Output

Submit Code

```

1  #!/usr/bin/env python
2
3  # Import the necessary modules
4  import sys
5  import os
6  import random
7  import time
8  import math
9
10 # Define the main function
11 def main():
12     # Get the number of iterations
13     iterations = int(sys.argv[1])
14
15     # Get the number of processors
16     processors = int(sys.argv[2])
17
18     # Get the number of tasks
19     tasks = int(sys.argv[3])
20
21     # Get the number of tasks per processor
22     tasks_per_processor = tasks // processors
23
24     # Get the number of tasks per processor
25     tasks_per_processor = tasks // processors
26
27     # Get the number of tasks per processor
28     tasks_per_processor = tasks // processors
29
30     # Get the number of tasks per processor
31     tasks_per_processor = tasks // processors
32
33     # Get the number of tasks per processor
34     tasks_per_processor = tasks // processors
35
36     # Get the number of tasks per processor
37     tasks_per_processor = tasks // processors
38
39     # Get the number of tasks per processor
40     tasks_per_processor = tasks // processors
41
42     # Get the number of tasks per processor
43     tasks_per_processor = tasks // processors
44
45     # Get the number of tasks per processor
46     tasks_per_processor = tasks // processors
47
48     # Get the number of tasks per processor
49     tasks_per_processor = tasks // processors
50
51     # Get the number of tasks per processor
52     tasks_per_processor = tasks // processors
53
54     # Get the number of tasks per processor
55     tasks_per_processor = tasks // processors
56
57     # Get the number of tasks per processor
58     tasks_per_processor = tasks // processors
59
60     # Get the number of tasks per processor
61     tasks_per_processor = tasks // processors
62
63     # Get the number of tasks per processor
64     tasks_per_processor = tasks // processors
65
66     # Get the number of tasks per processor
67     tasks_per_processor = tasks // processors
68
69     # Get the number of tasks per processor
70     tasks_per_processor = tasks // processors
71
72     # Get the number of tasks per processor
73     tasks_per_processor = tasks // processors
74
75     # Get the number of tasks per processor
76     tasks_per_processor = tasks // processors
77
78     # Get the number of tasks per processor
79     tasks_per_processor = tasks // processors
80
81     # Get the number of tasks per processor
82     tasks_per_processor = tasks // processors
83
84     # Get the number of tasks per processor
85     tasks_per_processor = tasks // processors
86
87     # Get the number of tasks per processor
88     tasks_per_processor = tasks // processors
89
90     # Get the number of tasks per processor
91     tasks_per_processor = tasks // processors
92
93     # Get the number of tasks per processor
94     tasks_per_processor = tasks // processors
95
96     # Get the number of tasks per processor
97     tasks_per_processor = tasks // processors
98
99     # Get the number of tasks per processor
100    tasks_per_processor = tasks // processors

```

Run or submit code when you're ready.