Output Window

Problem Solved Successfully ✓

Suggest Feedback

Test Cases Passed	Attempts : Correct / Total
1115 / 1115	**1 / 1**
	Accuracy : **100%**

Points Scored ⓘ	Time Taken
4 / 4	**0.84**
Your Total Score: 96 ↑	

Solve Next

Validate an IP Address   Multiply two strings   Implement Atoi

Stay Ahead With:

```java
class Solution {
    public int countBalanced(String[] arr) {
        // code here
        Map<Integer,Integer> hm = new HashMap<>();
        int sum = 0;
        int ans = 0;

        hm.put(0,1);
        for(String str : arr){
            int v = 0, c = 0;

            for(int i = 0; i < str.length(); i++){
                if(str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i'
                || str.charAt(i) == 'o' || str.charAt(i) == 'u') v++;
                else c++;
            }

            sum += (v-c);
            if(hm.containsKey(sum)) ans += hm.get(sum);

            hm.put(sum,hm.getOrDefault(sum,0)+1);
        }

        return ans;
    }
}
```

Ctrl + Enter

Custom Input   Compile & Run   Submit

Search...

Courses ∨   Tutorials ∨   Practice ∨   Jobs ∨

</> Problem   📄 Editorial   🕐 Submissions   💬 Comments

Java (21) ▾   ⏱ Start Timer ▶

```java
1  class Solution {
2      public int count(String s) {
3          // code here
4
5          HashMap<Character,Integer> map = new HashMap<>();
6
7          for(Character c : s.toCharArray() ){
8              map.put(c, map.getOrDefault(c, 0) + 1);
9          }
10
11         int count =0;
12         for(int c : map.values()){
13             if(c % 2 ==0){
14                 count++;
15             }
16         }
17
18         return count;
19     }
20 }
```

Output Window   _ ✕

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

Problem Solved Successfully ✓   Suggest Feedback

Test Cases Passed	Attempts : Correct / Total
1115 / 1115	**1 / 8**
	Accuracy : 12%

Points Scored ⓘ	Time Taken
2 / 2	**0.5**
Your Total Score: 92 ↑	

Solve Next

Run Length Encoding   Match specific pattern   Repetition of k length substring

Stay Ahead With:

Ctrl + Enter

Custom Input   Compile & Run   Submit

⠛⠛ | Search...

2 Offers Ending
Courses ⌄   Tutorials ⌄   Practice ⌄   Jobs ⌄   ᴬ ☾ 🔔 S

☰   </> Problem   📄 Editorial   🕐 Submissions   💬 Comments   Java (21) ⌄   ⏱ Start Timer ▶   ⧉ ⎗ ⚙ ↻ ⤢

Output Window   — ✕

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

Problem Solved Successfully ✔   Suggest Feedback

Test Cases Passed	Attempts : Correct / Total
102 / 102	**1 / 1**
	Accuracy : 100%

Points Scored ℹ	Time Taken
4 / 4	**0.11**
Your Total Score: 90 ↑	

Solve Next

Validate an IP Address   Multiply two strings   Implement Atoi

Stay Ahead With:

```java
1  class Solution {
2      public int vowelCount(String s) {
3          // code here
4          int n = s.length();
5          int ans = 1;
6          Map<Character,Integer> map = new HashMap<>();
7
8          for(int i = 0; i < n; i++){
9              char ch = s.charAt(i);
10             if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u'){
11                 map.put(ch,map.getOrDefault(ch,0)+1);
12             }
13         }
14
15         for(char ch : map.keySet()) ans *= map.get(ch);
16
17         return ans*fact(map.size());
18     }
19     private int fact(int n){
20         if(n < 3) return n;
21         if(n == 3) return 6;
22         if(n == 4) return 24;
23         if(n == 5) return 120;
24
25         return n;
26     }
27 };
```

💡   Custom Input   Compile & Run   Submit

Ctrl + Enter

</> Problem   📄 Editorial   🕐 Submissions   💬 Comments

Java (21) ▾   ⏱ Start Timer ⊙

Output Window   _ ✕

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

Problem Solved Successfully ✅

Suggest Feedback

Test Cases Passed   Attempts : Correct / Total

1120 / 1120   **1 / 1**

Accuracy : **100%**

Points Scored ⓘ   Time Taken

4 / 4   **0.53**

Your Total Score: 86 ↑

Solve Next

Majority Element   Fractional Knapsack   Minimum Platforms

Stay Ahead With:

```java
1  class Solution {
2      public static String maxSubseq(String s, int k) {
3          // code here
4          int n = s.length();
5          Stack<Character> stack = new Stack<>();
6          String result = "";
7          for(int i=0; i<n; i++){
8              char ch = s.charAt(i);
9              while(!stack.isEmpty() && k>0 && stack.peek()<ch){
10                 stack.pop();
11                 k--;
12             }
13             stack.push(ch);
14         }
15         while(!stack.isEmpty() && k>0){
16             stack.pop();
17             k--;
18         }
19         while(!stack.isEmpty()){
20             result = stack.pop() + result;
21         }
22         return result;
23     }
24 };
```

Ctrl + Enter

Custom Input   Compile & Run   Submit

Search...

Courses ⌄ | Tutorials ⌄ | Practice ⌄ | Jobs ⌄

</> Problem | 🗎 Editorial | ⏱ Submissions | 💬 Comments

Java (21) ⌄ | ⏱ Start Timer ⏵

Output Window — ✕

Compilation Results | Custom Input | Y.O.G.I. (AI Bot)

Problem Solved Successfully ✔

Suggest Feedback

Test Cases Passed

1110 / 1110

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored ℹ

2 / 2

Your Total Score: 82 ↑

Time Taken

0.98

Solve Next

Run Length Encoding | Match specific pattern | Repetition of k length substring

Stay Ahead With:

```java
5    String key = "";
6    HashMap<String , Integer> map = new HashMap<>();
7
8    for(int i = 0 ; i<arr.size() ; i++){
9        map.put(arr.get(i) , map.getOrDefault(arr.get(i) , 0)+1);
10   }
11
12   for(Map.Entry<String , Integer> entry: map.entrySet()){
13       String s = entry.getKey();
14       int freq = entry.getValue();
15
16       if(freq >targetValue){
17           targetValue = freq;
18
19           key = s;
20       }
21
22   }
23       for(Map.Entry<String , Integer> entr: map.entrySet()){
24           String st = entr.getKey();
25           int fre = entr.getValue();
26           if(fre>=targetValue){
27               int comparison = key.compareTo(st);
28               if(comparison>0){
29                   key = st;
30               }
31           }
32
33       }
34   ArrayList<String> str = new ArrayList<>();
35
36   str.add(key);
37   str.add(Integer.toString(targetValue));
38   return str;
39
40   }
41 };
```

Ctrl + Enter

Custom Input | Compile & Run | Submit

Output Window   _ ✕

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

Problem Solved Successfully ✔   Suggest Feedback

Test Cases Passed	Attempts : Correct / Total
1120 / 1120	**1 / 4**
	Accuracy : 25%

Points Scored ⓘ	Time Taken
4 / 4	**3.16**
Your Total Score: 80 ↑	

Solve Next

[Validate an IP Address]   [Multiply two strings]   [Implement Atoi]

Stay Ahead With:

```java
1  class Solution {
2      public static int countPairs(String[] arr) {
3          int count = 0;
4          int n = arr.length;
5
6          for (int i = 0; i < n; i++) {
7              for (int j = i + 1; j < n; j++) {
8                  if (differByOne(arr[i], arr[j])) {
9                      count++;
10                 }
11             }
12         }
13
14         return count;
15     }
16
17     public  static boolean differByOne(String a, String b) {
18         int mismatch = 0;
19         for (int i = 0; i < a.length(); i++) {
20             if (a.charAt(i) != b.charAt(i)) {
21                 mismatch++;
22                 if (mismatch > 1) return false;
23             }
24         }
25         return mismatch == 1;
26     }
27 };
```

Ctrl + Enter

Custom Input   Compile & Run   Submit

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

```
1  class Solution {
2      public void sortByLength(String[] arr) {
3          // code here
4          Arrays.sort(arr,Comparator.comparingInt(String::length));
5      }
6  }
```

Problem Solved Successfully ✔   Suggest Feedback

Test Cases Passed   Attempts : Correct / Total

1111 / 1111   You can see all your attempts in submission tab

                                           Accuracy : **100%**

Points Scored ⓘ   Time Taken

You can see the score in submission tab   **1.06**

⚠ Calculating score...

Solve Next

Validate an IP Address   Multiply two strings   Implement Atoi

☀   Custom Input   Compile & Run   Submit

Ctrl + Enter

Output Window

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

Problem Solved Successfully ✅

Suggest Feedback

Test Cases Passed	Attempts : Correct / Total
1117 / 1117	**1 / 3**
	Accuracy : **33%**
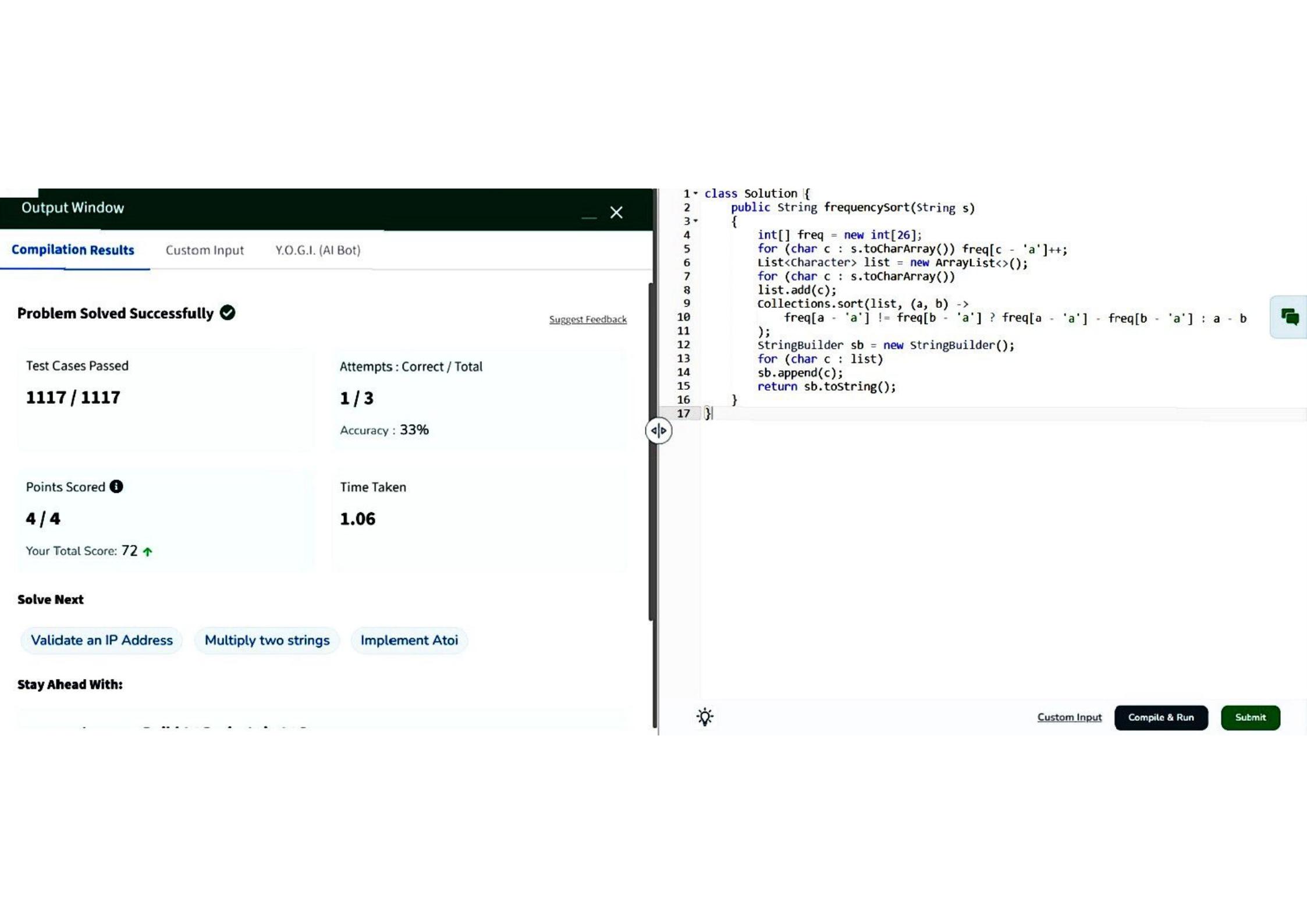
Points Scored ℹ️	Time Taken
4 / 4	**1.06**
Your Total Score: 72 ⬆	

Solve Next

Validate an IP Address   Multiply two strings   Implement Atoi

Stay Ahead With:

```java
1  class Solution {
2      public String frequencySort(String s)
3      {
4          int[] freq = new int[26];
5          for (char c : s.toCharArray()) freq[c - 'a']++;
6          List<Character> list = new ArrayList<>();
7          for (char c : s.toCharArray())
8          list.add(c);
9          Collections.sort(list, (a, b) ->
10             freq[a - 'a'] != freq[b - 'a'] ? freq[a - 'a'] - freq[b - 'a'] : a - b
11         );
12         StringBuilder sb = new StringBuilder();
13         for (char c : list)
14         sb.append(c);
15         return sb.toString();
16     }
17 }
```

Custom Input   Compile & Run   Submit

Output Window

Problem Solved Successfully ✅

Test Cases Passed	Attempts : Correct / Total
1120 / 1120	**1 / 1**
	Accuracy : 100%

Points Scored ⓘ	Time Taken
4 / 4	**1.42**

Your Total Score: 68 ↑

Solve Next

Validate an IP Address Multiply two strings Implement Atoi

Stay Ahead With:

```java
class Solution {
    public int minDifference(String[] arr) {
        // code here
        int [] timeInSeconds = new int[arr.length];
        for(int i = 0;i<arr.length;i++)
        {
            String[] parts = arr[i].split(":");
            int hours = Integer.parseInt(parts[0]);
            int minutes = Integer.parseInt(parts[1]);
            int seconds = Integer.parseInt(parts[2]);
            timeInSeconds[i] = hours * 3600 + minutes * 60 + seconds;
        }
        Arrays.sort(timeInSeconds);
        int minDiff = Integer.MAX_VALUE;
        int n = timeInSeconds.length;

        for(int i = 0; i<n-1; i++){
            int diff = timeInSeconds[i+1]-timeInSeconds[i];
            minDiff = Math.min(minDiff,diff);
        }
        int lastToFirst = (86400 - timeInSeconds[n - 1])+timeInSeconds[0];
        minDiff = Math.min(minDiff,lastToFirst);
        return minDiff;

    }
}
```

Ctrl + Enter

Custom Input Compile & Run Submit

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

Problem Solved Successfully ✓

Suggest Feedback

Test Cases Passed

1120 / 1120

Attempts : Correct / Total

1 / 2

Accuracy : **50%**

Points Scored ⓘ

4 / 4

Your Total Score: **64** ↑

Time Taken

0.07

Solve Next

The Celebrity Problem   Get Min from Stack   Histogram Max Rectangular Area

Stay Ahead With:

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

Problem Solved Successfully ✔

Suggest Feedback

Test Cases Passed

1111 / 1111

Attempts : Correct / Total

1 / 2

Accuracy : **50%**

Points Scored ℹ

4 / 4

Time Taken

0.17

Your Total Score: **60** ↑

Solve Next

The Celebrity Problem   Get Min from Stack   Histogram Max Rectangular Area

Stay Ahead With:

Compilation Results   Custom Input   Y.O.G.I. (AI Bot)

Problem Solved Successfully ✓

Suggest Feedback

Test Cases Passed	Attempts : Correct / Total
1120 / 1120	**1 / 1**
	Accuracy : 100%

Points Scored ⓘ	Time Taken
2 / 2	**0.48**
Your Total Score: 56 ↑	

Solve Next

Huffman Decoding   Huffman Decoding-1   N meetings in one room

Stay Ahead With:

```java
1  class Solution {
2      public int minSwaps(String s1, String s2) {
3          // code here
4          int count1 = 0;
5          int count2 = 0;
6
7          for(int i=0;i<s1.length();i++){
8              if(s1.charAt(i)=='0'&& s2.charAt(i)=='1'){
9                  count1++;
10             }
11             if(s1.charAt(i)=='1' && s2.charAt(i)=='0'){
12                 count2++;
13             }
14         }
15
16         int res = (count1/2)+(count2/2);
17         if(count1%2==0 && count2%2==0){
18             return res;
19         }
20         else if((count1+count2)%2==0){
21             return res+2;
22         }
23         return -1;
24     }
25 }
26
27
```

Ctrl + Enter

Custom Input   Compile & Run   Submit