

 \sim

Q

 \Box





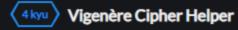


VIM EMACS



SUBMIT





Instructions Output

You have passed all of the tests! :)

Time: 869ms Passed: 4 Failed: 0 Test Results: → Testing with lowercase latin alphabet > Password of 'password' > Password of 'pizza' > Password of 'attackatdawn', random tokens Completed in 2ms → Testing with katakana alphabet > Password of 'カタカナ'

☆ 785 🕏 130 🛷 90% of 974 📵 3,029 of 7,197 💄 jacobb 🛕 7 Issues Reported

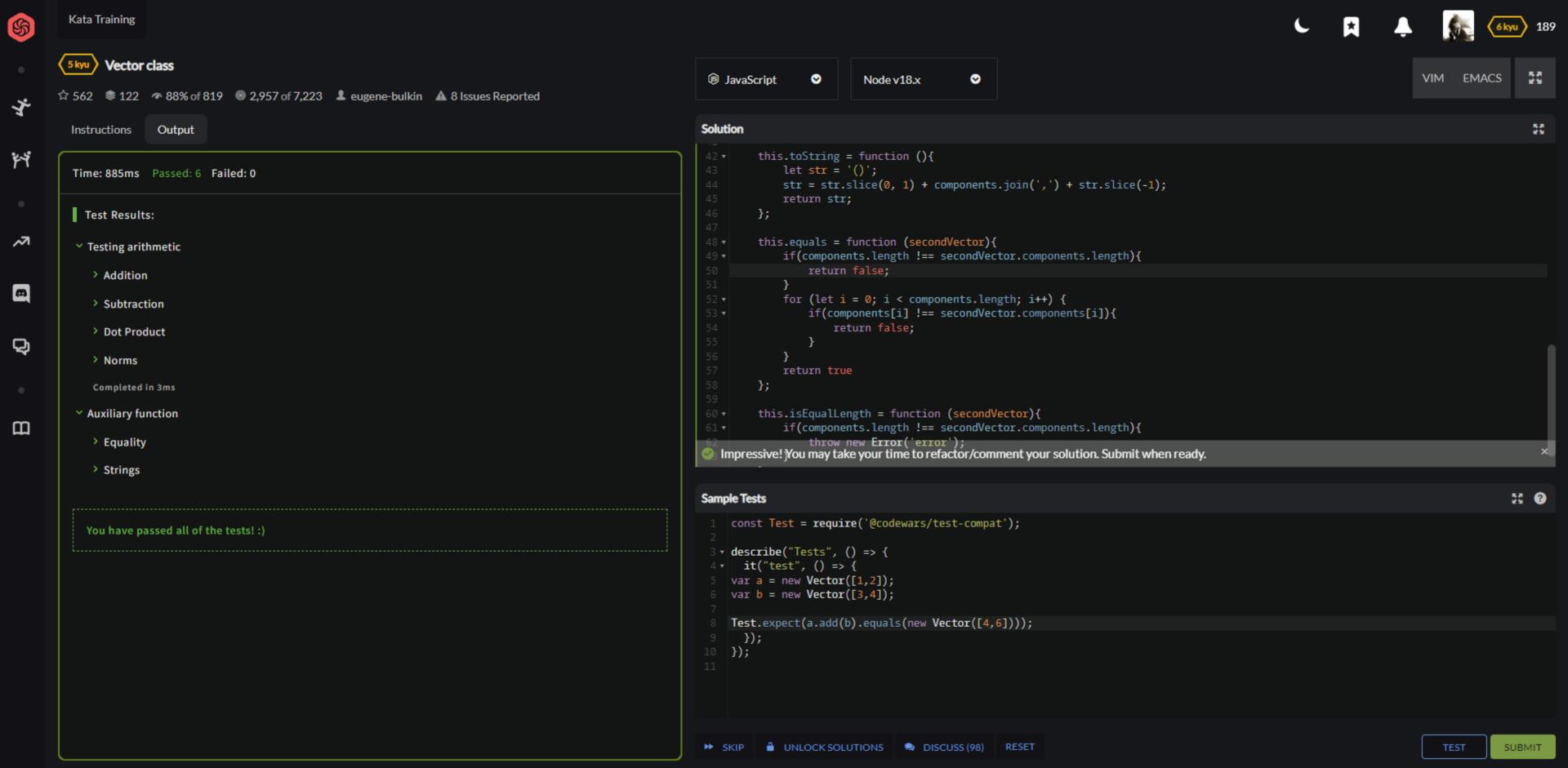
```
    JavaScript

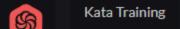
                                                     \odot
                               Node v18.x
```

▶ SKIP DISCUSS (156) RESET

```
Solution
             for (let i = 0; i < str.length; i++) {
                // якщо символу в алфавіті немає - додати символ з даного рядка
                if(abc.index0f(str[i]) === -1){
                    cipher += str[i];
                else{ // додати символ, який знаходиться в алфавіті на відстані відповідного символу з кеу від символу із заданого рядк
                    cipher += abc.slice((abc.indexOf(str[i]) - abc.indexOf(newKey[i])) % abc.length, ((abc.indexOf(str[i]) - abc.indexO
            return cipher;
         };
         // створення ключа, який буде довжиною як задана строка
         this.createKey = function (str){
            const concatenationNum = Math.floor(str.length / key.length);
            const sliceNum = str.length % key.length;
            let newKey = key.repeat(concatenationNum);
            newKey += key.slice(0, sliceNum);
            return newKey;
Good Job! You may take your time to refactor/comment your solution. Submit when ready.
```

```
33 3
Sample Tests
   const Test = require('@codewars/test-compat');
  describe("Tests", () => {
 4 it("test", () => {
   var abc, key;
    abc = "abcdefghijklmnopqrstuvwxyz";
    key = "password"
   c = new VigenèreCipher(key, abc);
    Test.assertEquals(c.encode('codewars'), 'rovwsoiv');
    Test.assertEquals(c.decode('rovwsoiv'), 'codewars');
    Test.assertEquals(c.encode('waffles'), 'laxxhsj');
   Test.assertEquals(c.decode('laxxhsj'), 'waffles');
```

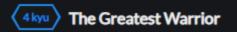




 \sim

Q

Ш



🔯 645 🏶 133 🛷 93% of 564 📵 810 of 2.987 🎩 BurstNova 🛕 6 Issues Reported Output Instructions

Time: 739ms Passed: 2 Failed: 0 Test Results: Kata Tests > Standard Tests > Random Tests Completed in 4ms You have passed all of the tests!:)

```
    JavaScript

                  <
                            Node v18.x
```

DISCUSS (74) RESET

VIM EMACS 0

```
Solution
        battle = (lvl) => {
          if(lvl < 1 || lvl > 100){
                 return "Invalid level":
            if(((lvl - this.level()) >= 5) && (Math.floor(this.level()/10) < Math.floor(lvl/10))){</pre>
                 return "You've been defeated";
            if(lvl === this.level()){
                this.expAdd(10):
                return "A good fight";
            if((this.level() - lvl) === 1){
                 this.expAdd(5);
                return "A good fight";
            if((this.level() - lvl) > 1){
                 return "Easy fight";
            if((lvl - this.level()) > 0){
                 this.expAdd(20 * Math.abs(lvl - this.level()) * Math.abs(lvl - this.level()));
                 return "An intense fight":
Correct! You may take your time to refactor/comment your solution. Submit when ready.
```

Sample Tests

```
Test.assertDeepEquals(Goku.level(), 1, "All warriors must start as LV 1")
Test.assertDeepEquals(Goku.rank(), "Pushover", "All warriors must start as a Pushover")
Test.assertDeepEquals(Goku.achievements(), [], "All warriors must start off with no achievements to their name")
Test.assertDeepEquals(Goku.training(["Do ten push-ups", 95, 1]), "Do ten push-ups", "After obtaining a mastery, the achievement
Test.assertDeepEquals(Goku.level(), 1, "After your warriors's first training, he/she must still be LV 1")
Test.assertDeepEquals(Goku.battle(0), "Invalid level", "LV 0 enemies do not exist. This must be a mistake!")
Test.assertDeepEquals(Goku.battle(1), "A good fight", "After fighting against a warrior of equal level, your warrior should pro
Test.assertDeepEquals(Goku.level(), 2, "With that fight under your belt, your warrior should now be ascended to LV 2")
Test.assertDeepEquals(Goku.achievements(), ["Do ten push-ups"], "Don't forget what your warrior has achieved so far")
Test.assertDeepEquals(Goku.rank(), "Pushover", "A pushover, however, your warrior should remain")
Test.assertDeepEquals(Goku.battle(3), "An intense fight", "Another battle, this time with a slightly more skilled enemy, should
Test.assertDeepEquals(Goku.level(), 2, "Your should still remain LV 2")
Test.assertDeepEquals(Goku.training(["Survive one night at the Forest of Death", 170, 2]), "Survive one night at the Forest of
                                        stered the Spirit Bomb", 1580, 10]), "Not strong enough", "Your warrior isn't skilled
```

33 3

SUBMIT

