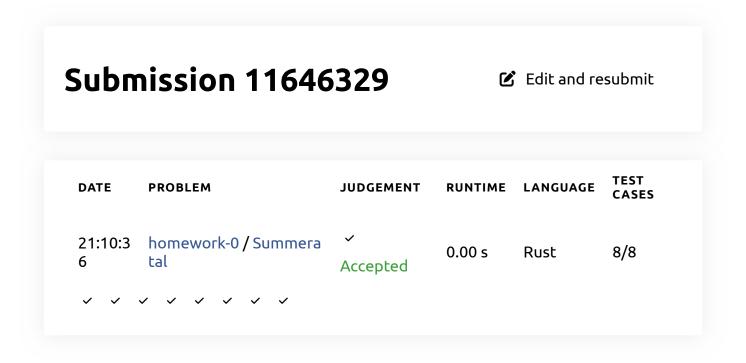
homework-0



Submission history (at most 10 last submissions)

1 av 3

DATE	JUDGEMENT	RUNTIME	LANGUAGE	DIFF	TEST CASES	
21:10: 36	✓ Accepted	0.00 s	Rust	Current submission	8/8	View Details
~ ~	~ ~ ~ ~	~ ~				
20:55: 01	✓ Accepted	0.00 s	Rust	Diff with this submission	8/8	View Details
~ ~	~ ~ ~ ~	~ ~				
20:41: 35	× Compile Error		Rust	Diff with this submission	-/-	View Details
20:39: 50	× Compile Error		Rust	Diff with this submission	-/-	View Details
20:35: 05	× Compile Error		Rust	Diff with this submission	-/-	View Details
17:45: 34	× Compile Error		Rust	Diff with this submission	-/-	View Details

Files submitted

2 av 3

kth_javap_sumsort.rs

▲ Download file

```
1 #![allow(dead code)]
2 #![allow(unused variables)]
3 #![allow(warnings)]
4
5 use std::any::Any;
6 use std::io;
7 use std::io::prelude::*;
8 use std::collections::BTreeSet;
9
10 fn main() {
11
       let input = io::stdin();
12
       let mut line1 = String::new();
13
       input.lock().read line(&mut line1).unwrap();
       let n = line1.trim().parse::<i32>().unwrap();
14
       let searches = if n % 2 == 0 \{ n/2 \} else \{ (n+1)/2 \};
15
16
       let mut line2 = String::new();
17
       input.lock().read_line(&mut line2).unwrap();
18
       let mut nums: Vec<i32> = Vec::new();
19
       line2.trim()
           .split(" ")
20
21
           .map(|_token| _token.parse::<i32>())
22
           .for_each(|_token| {
23
               if _token.is_ok() {
                    let i1 = _token.unwrap();
24
25
                    nums.push(i1);
26
               }
27
           });
28
       nums.sort_by(|_num1, _num2| _num1.cmp(_num2));
29
       let sum = nums.iter()
           .skip((n - searches) as usize)
30
           .map(|_x| *_x)
31
           .reduce(|_num1, _num2| _num2 + _num1)
32
33
           .unwrap();
34
       println!("{}", sum);
35 }
36
```

3 av 3