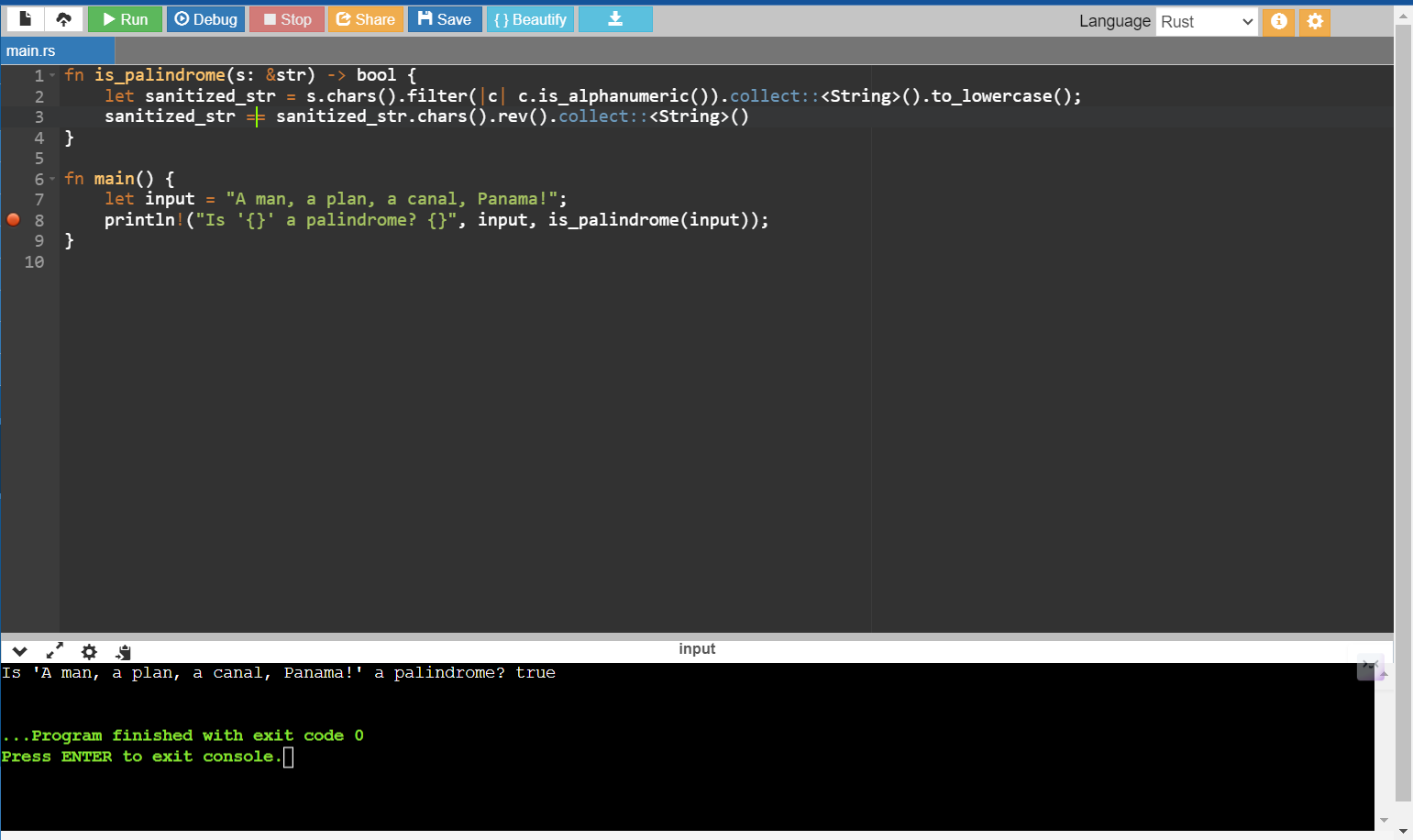
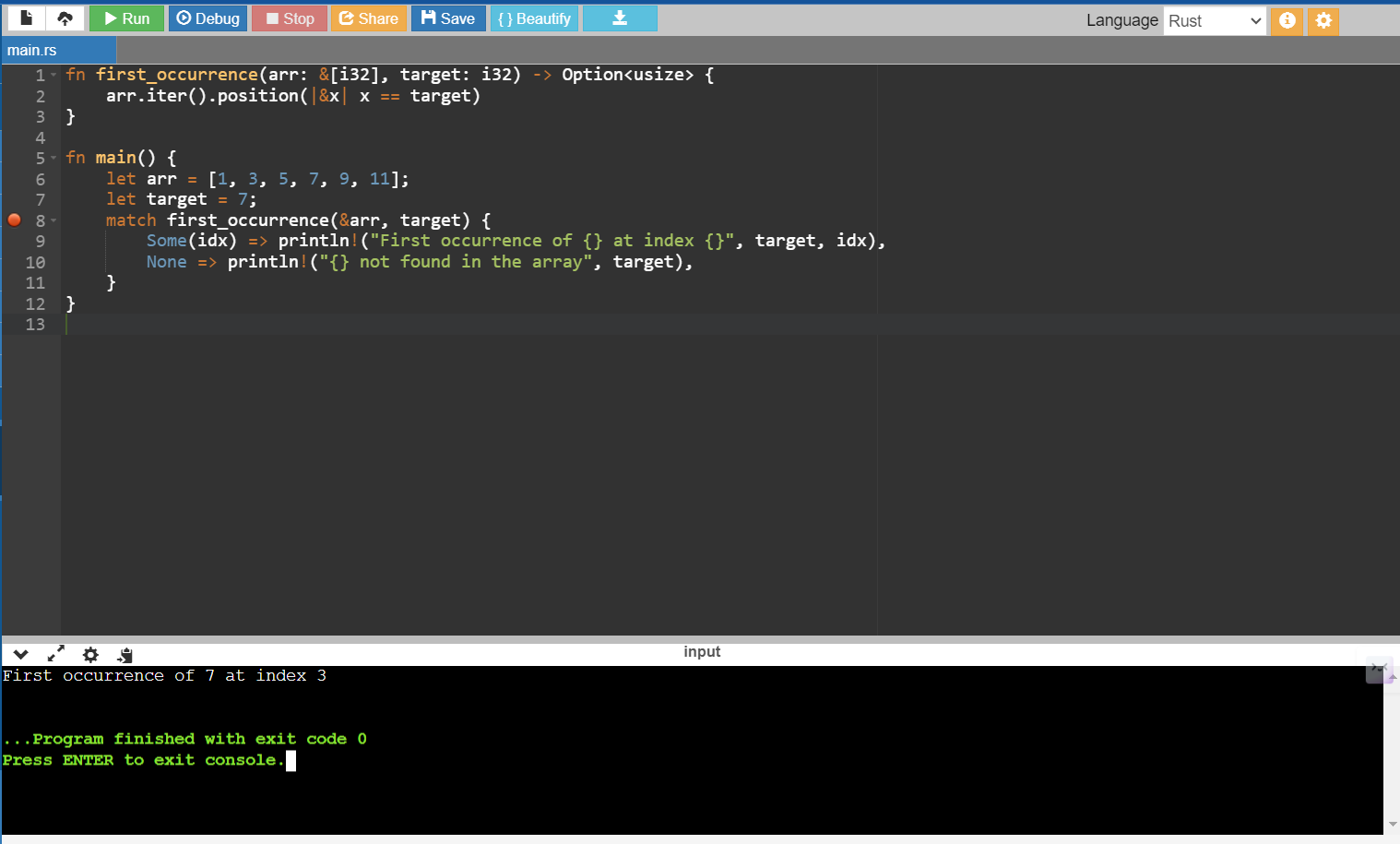
Ques 1.

Implement a function that checks whether a given string is a palindrome or not.



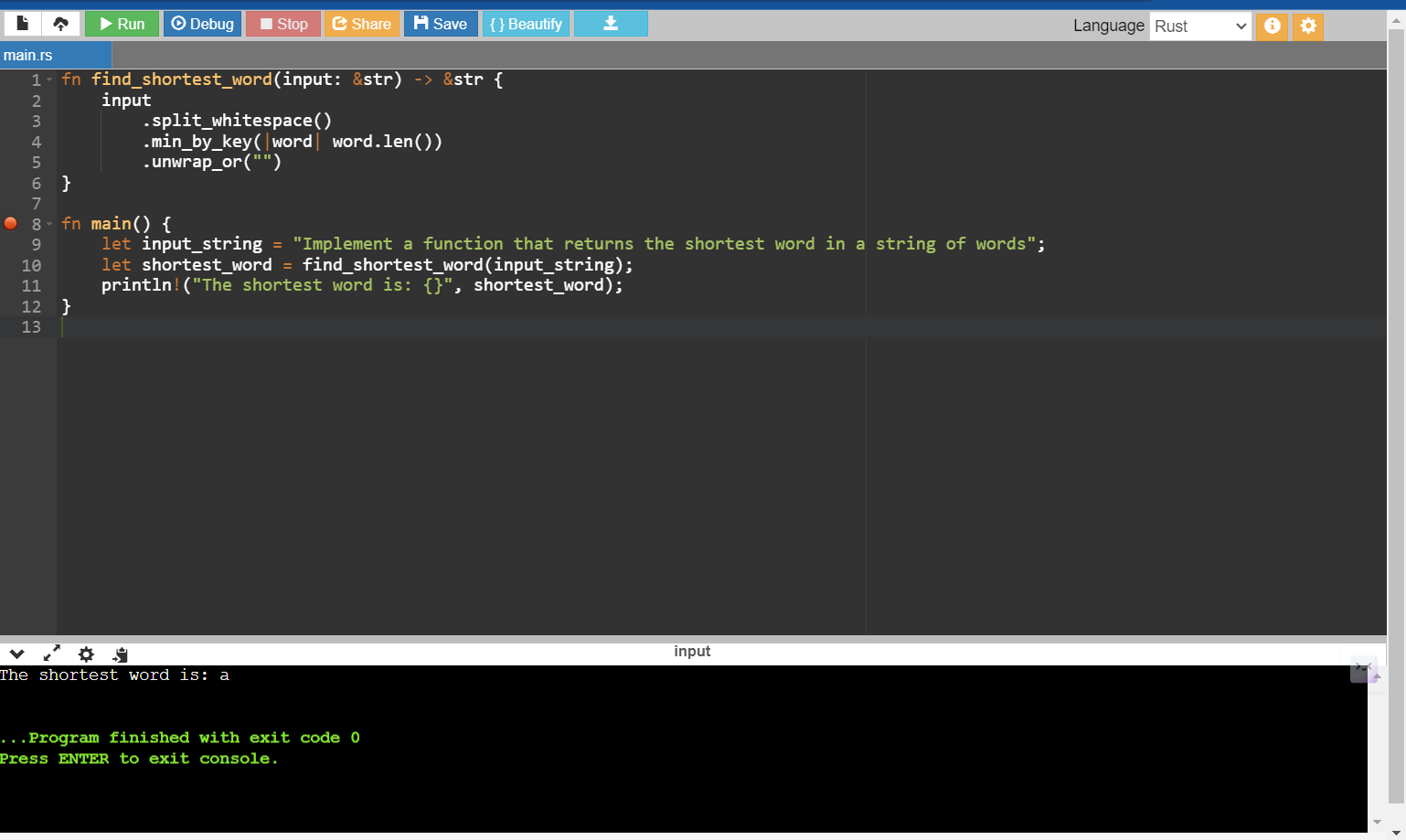
Ques 2

Given a sorted array of integers, implement a function that returns the index of the first occurrence of a given number



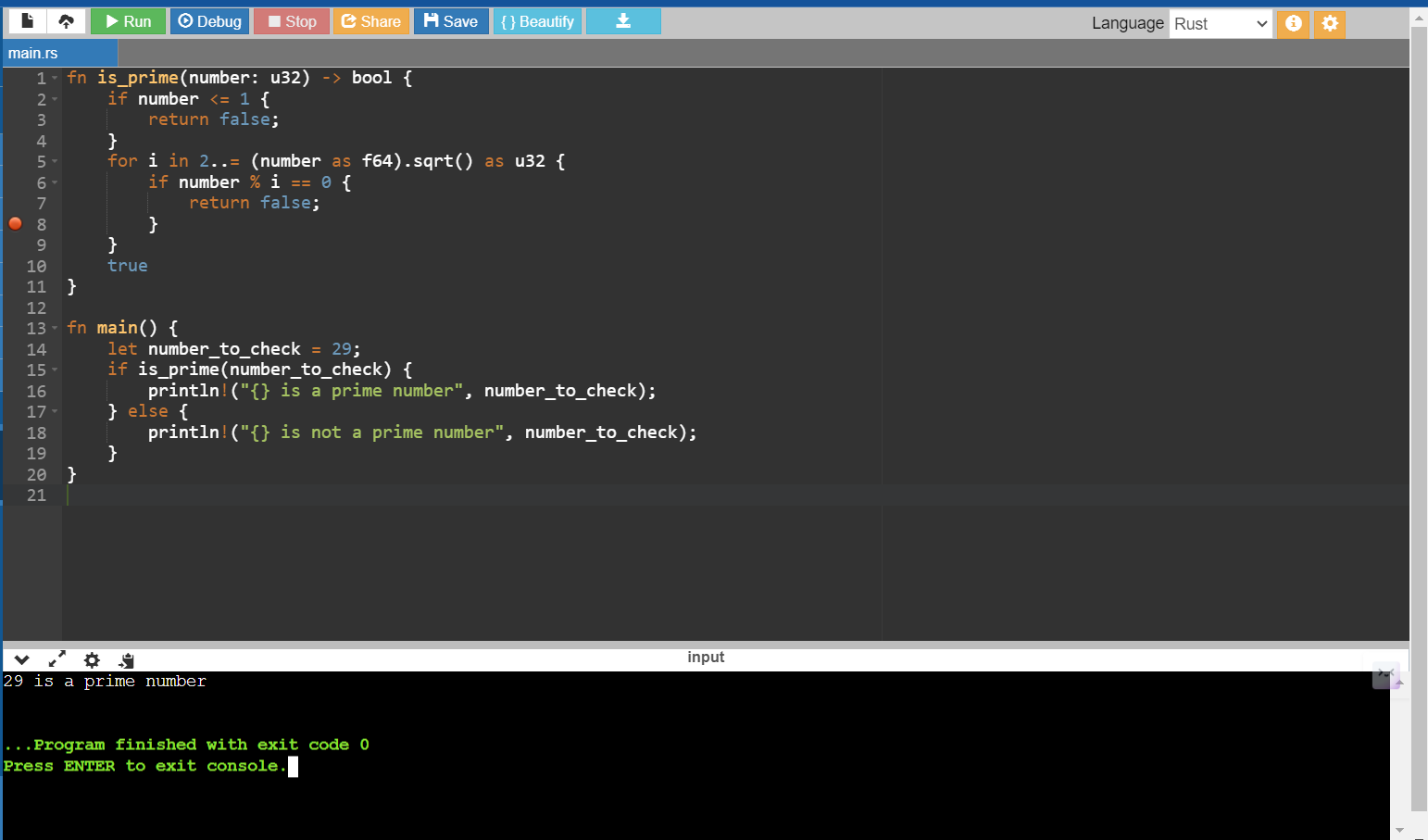
Ques 3

Given a string of words, implement a function that returns the shortest word in the string.



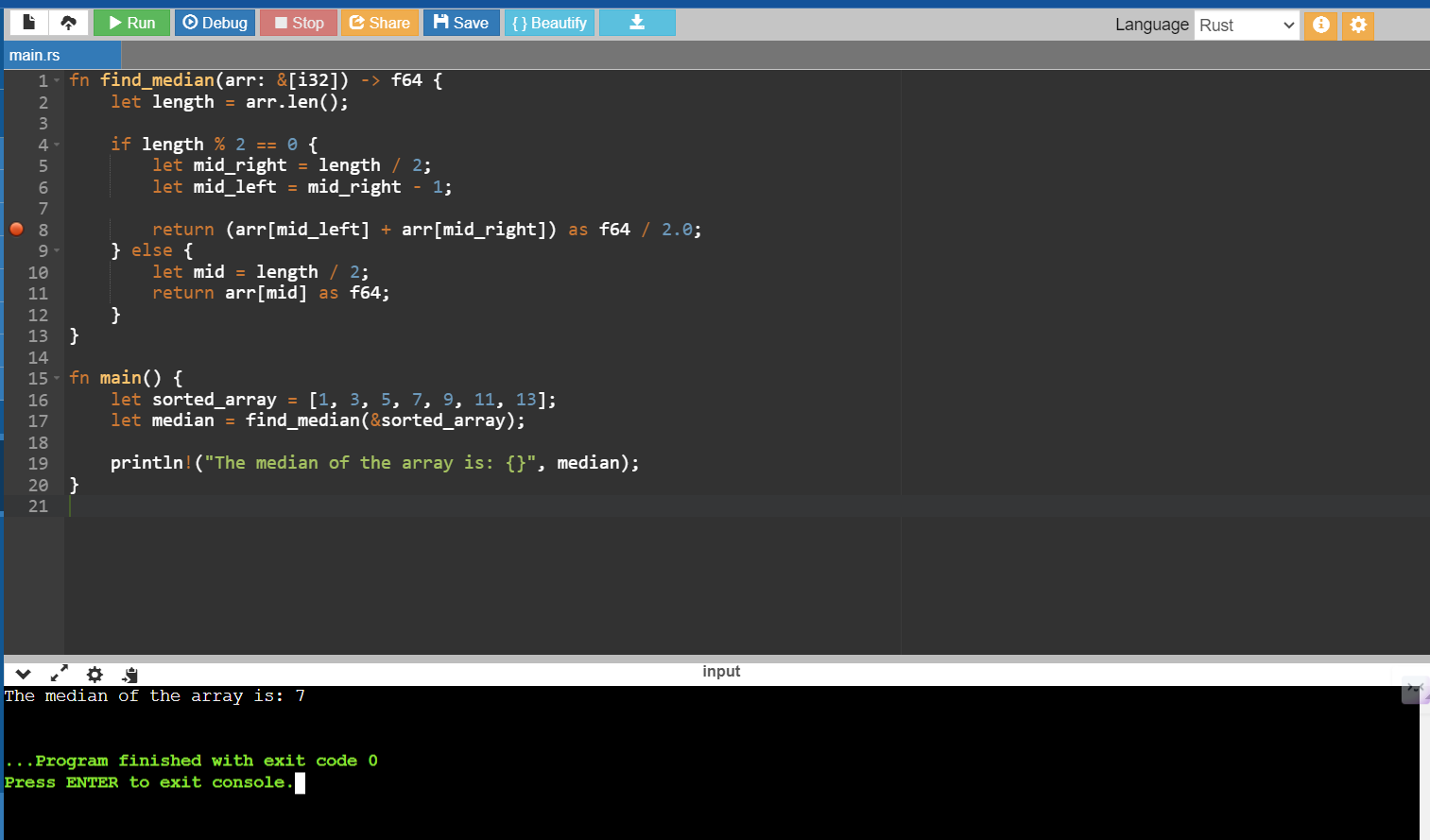
Quest 4

Implement a function that checks whether a given number is prime or not.



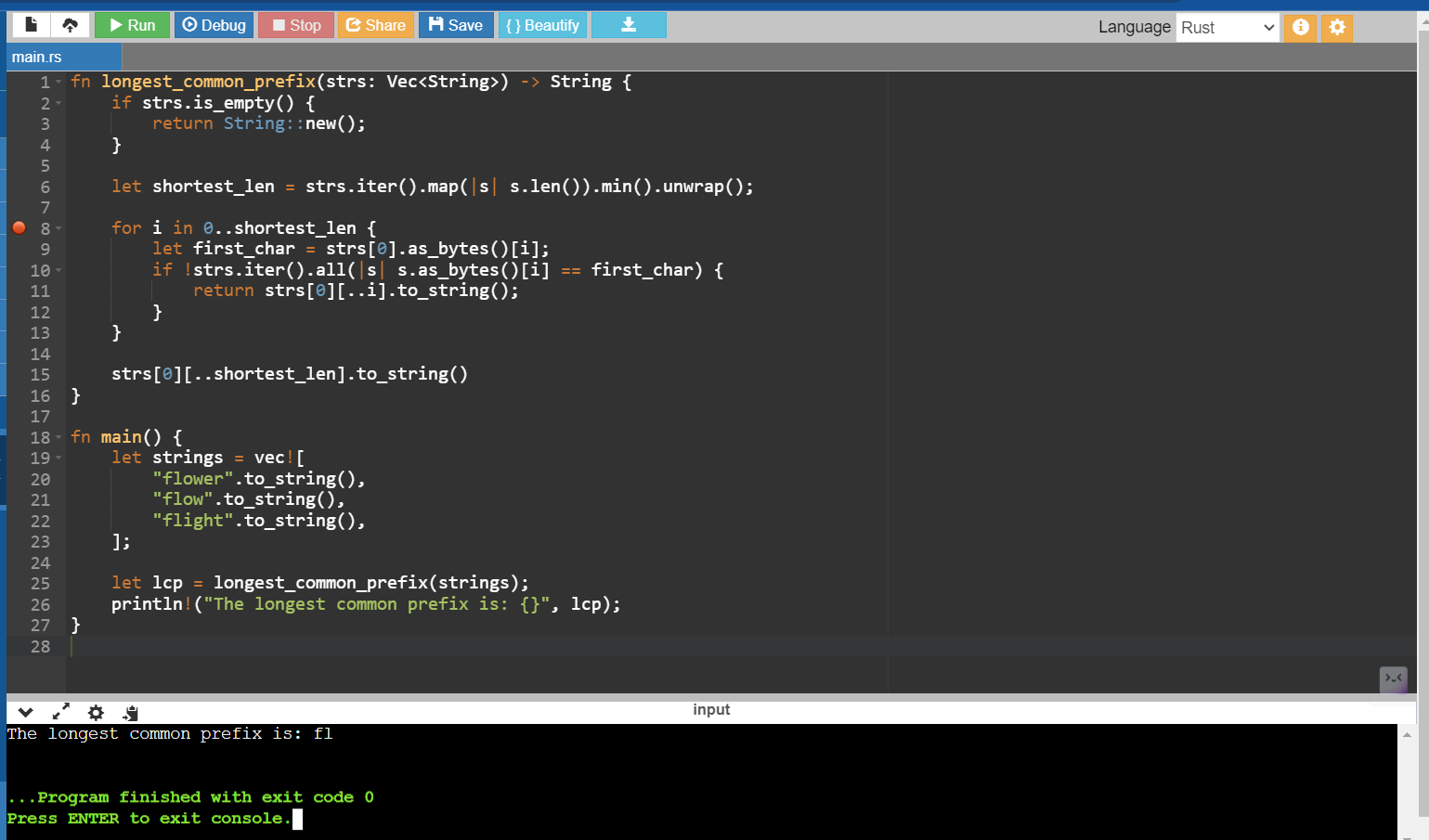
Ques 5

Given a sorted array of integers, implement a function that returns the median of the array.



Ques 6

Implement a function that finds the longest common prefix of a given set of strings.



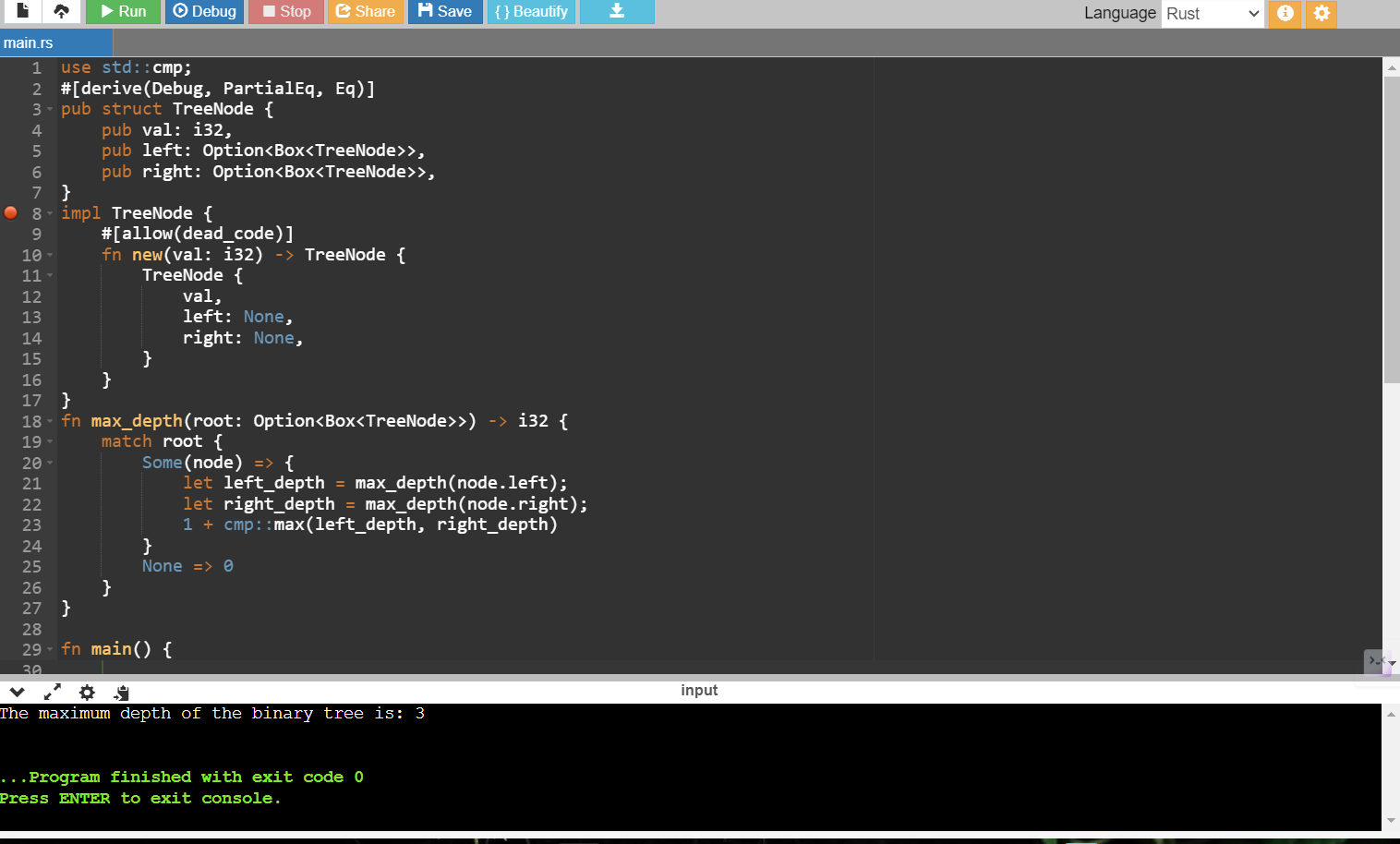
Ques 7

Implement a function that returns the kth smallest element in a given array.



Ques 8

Given a binary tree, implement a function that returns the maximum depth of the tree.



use std::cmp;

#[derive(Debug, PartialEq, Eq)]

pub struct TreeNode {

pub val: i32,

pub left: Option<Box<TreeNode>>,

pub right: Option<Box<TreeNode>>,

}

impl TreeNode {

#[allow(dead\_code)]

fn new(val: i32) -> TreeNode {

TreeNode {

val,

left: None,

right: None,

}

}

}

fn max\_depth(root: Option<Box<TreeNode>>) -> i32 {

match root {

Some(node) => {

let left\_depth = max\_depth(node.left);

let right\_depth = max\_depth(node.right);

1 + cmp::max(left\_depth, right\_depth)

}

None => 0

}

}

fn main() {

let tree = Some(Box::new(TreeNode {

val: 3,

left: Some(Box::new(TreeNode {

val: 9,

left: None,

right: None,

})),

right: Some(Box::new(TreeNode {

val: 20,

left: Some(Box::new(TreeNode {

val: 15,

left: None,

right: None,

})),

right: Some(Box::new(TreeNode {

val: 7,

left: None,

right: None,

})),

})),

}));

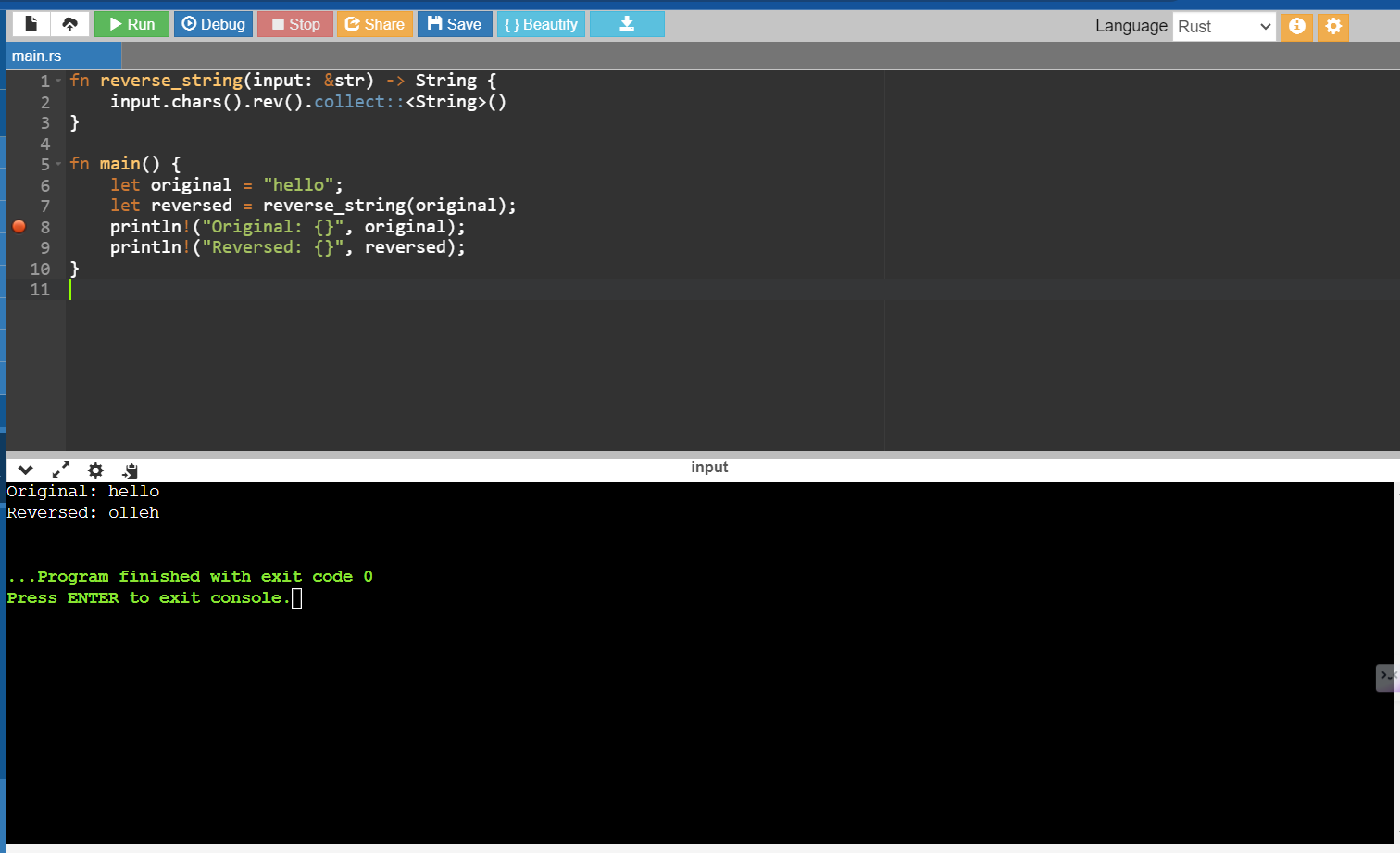
let depth = max\_depth(tree);

println!("The maximum depth of the binary tree is: {}", depth);

}

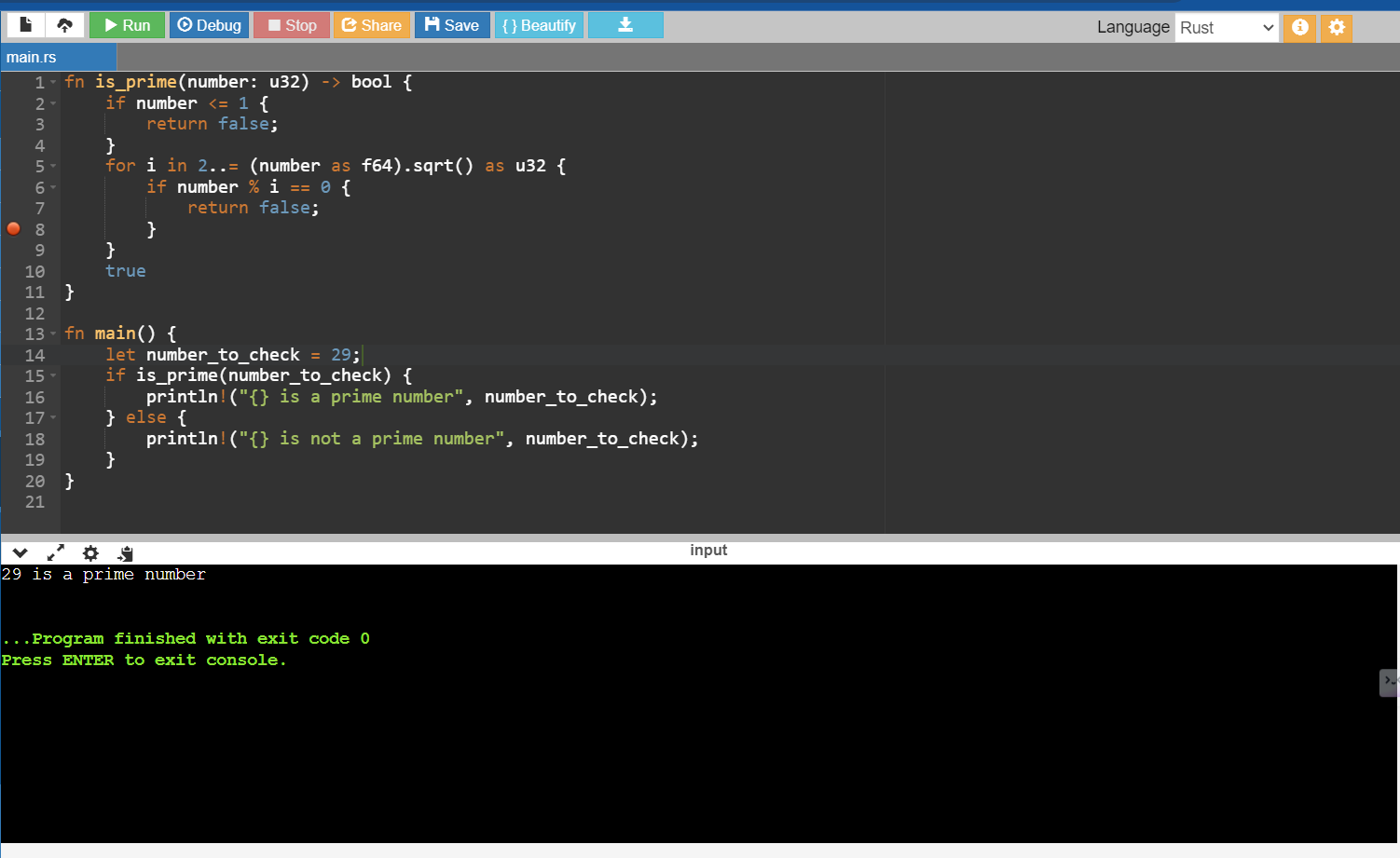
Ques 9

Reverse a string in Rust



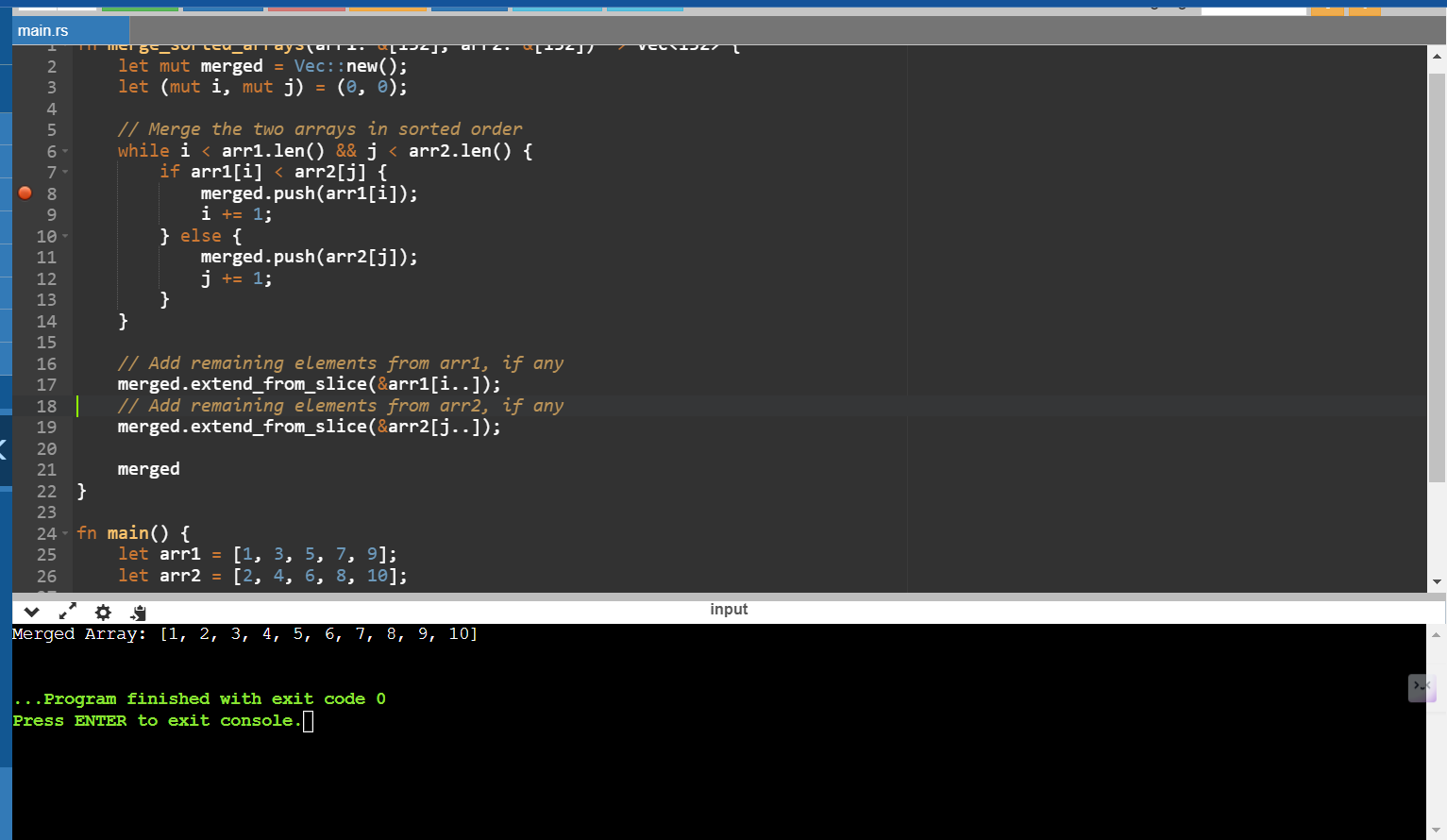
Ques 10

Check if a number is prime in Rust



Ques 11

Merge two sorted arrays in Rust



Ques 12

Find the maximum subarray sum in Rust

