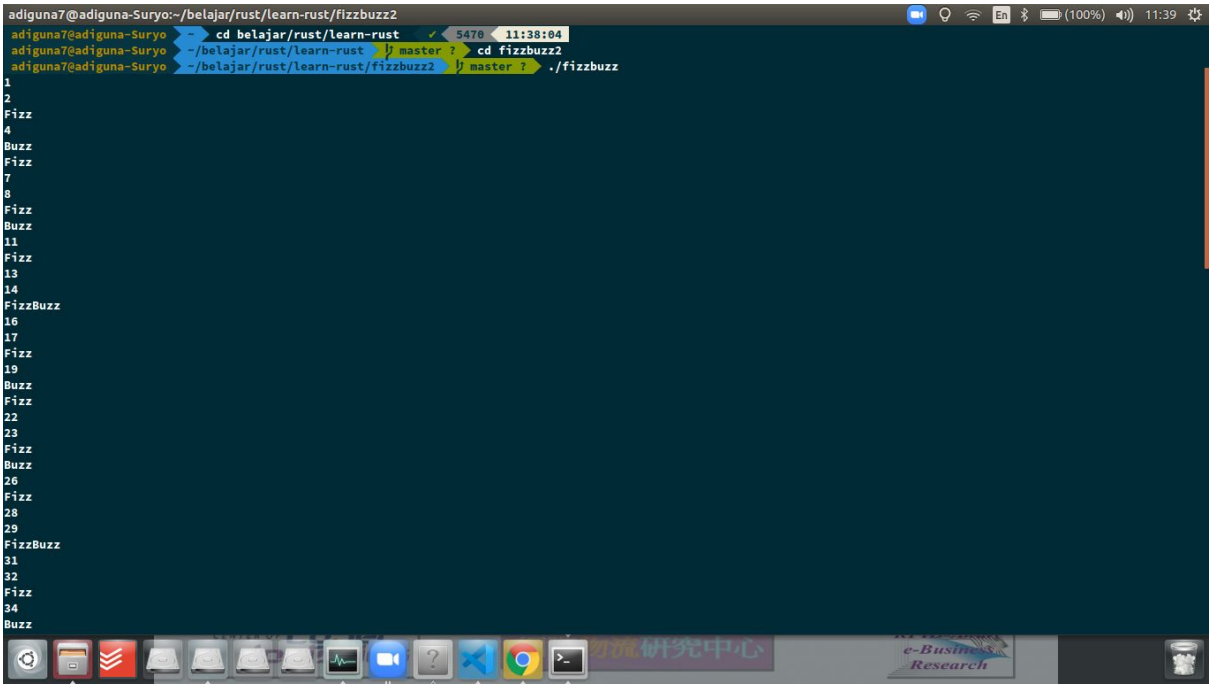
Tugas 1 Mobile Programming

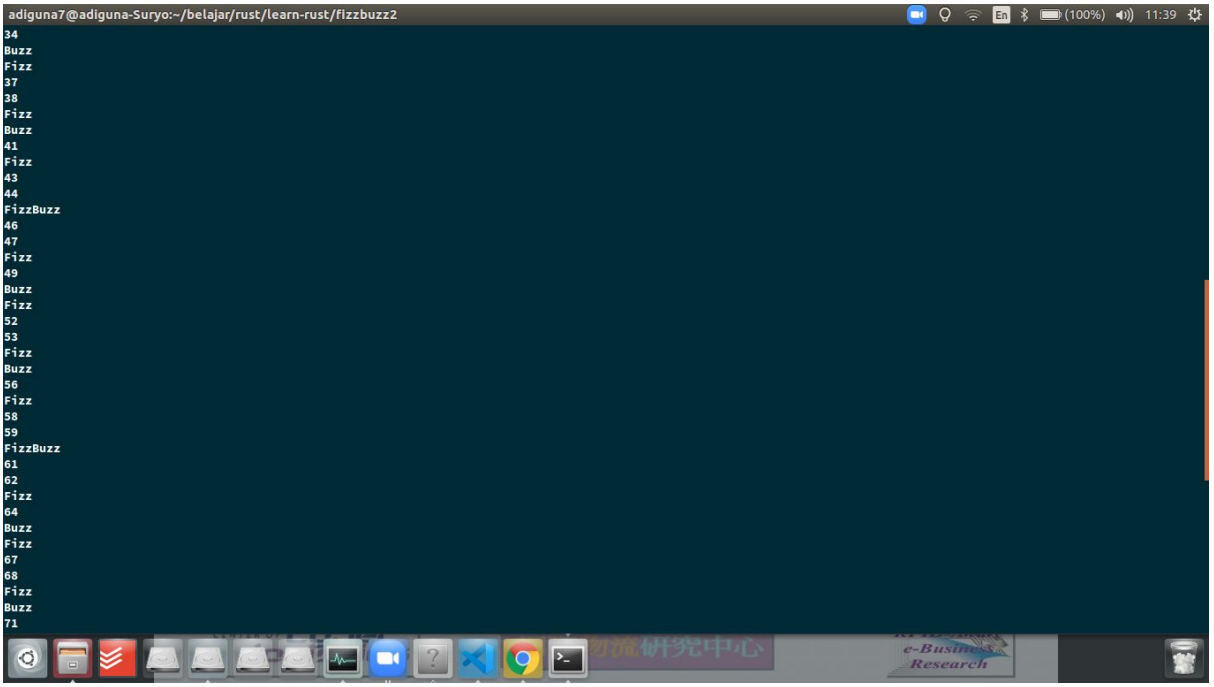
1. Fizzbuzz program
   (print angka 1 - 100, apabila angka tersebut dapat dibagi 3 maka print Fizz, apabila angka tersebut dapat dibagi 5 maka print Buzz. apabila dapat dibagi keduanya maka print FizzBuzz)

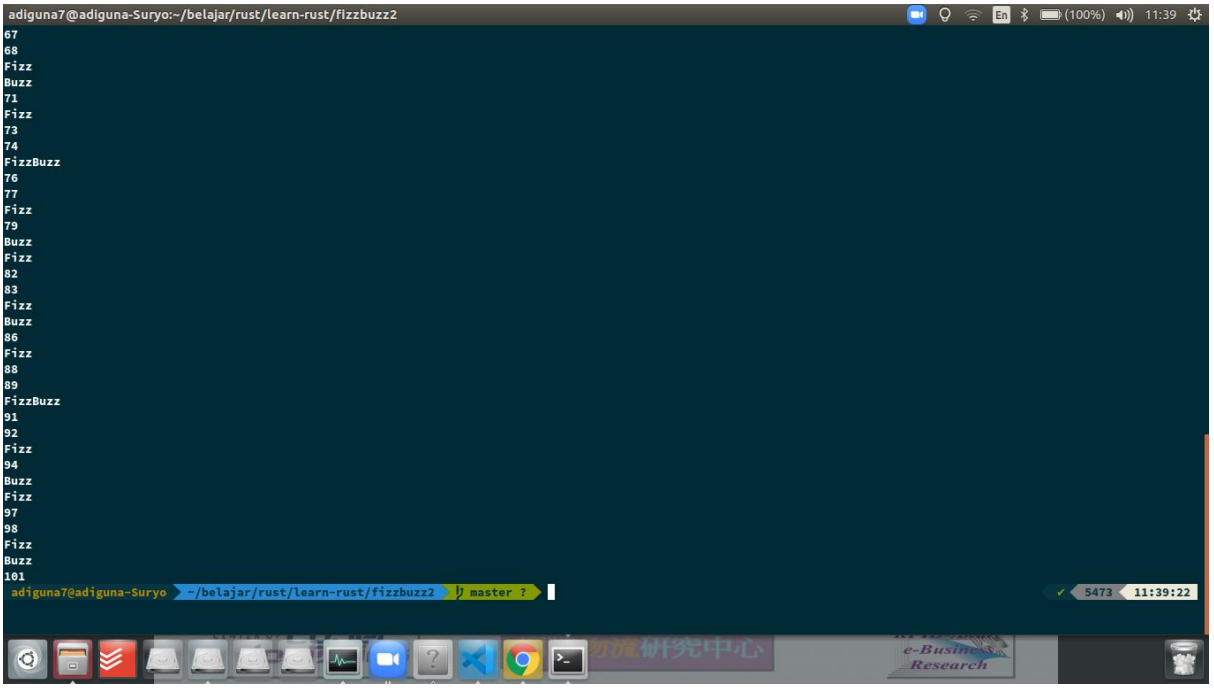
   Kode Program

```rust
fn main() {
    for i in 1..102 {
        match (i%3, i%5) {
            (0, 0) => println!("FizzBuzz"),
            (0, _) => println!("Fizz"),
            (_, 0) => println!("Buzz"),
            (_, _) => println!("{}", i)
        }
    }
}
```

   Output:







2. Oddman Out

   Mencari bilangan yang tidak memiliki pasangan pada array

   array input: [1, 2, 3, 4, 5, 1, 2, 3, 4]

   Kode Program:

```rust
pub fn odd_man_out(l: &[i32]) -> i32 {
    let mut mask : i32 = 0;
    for x in l {
        mask ^= *x;
    }
```

```
    return mask;
}

fn main(){
    print!("{}", odd_man_out(&[1, 2, 3, 4, 5, 1, 2, 3, 4]));
}
```

Output:

```
adiguna7@adiguna-Suryo  ~/belajar/rust/learn-rust/oddman2  master ?  rustc oddman.rs
adiguna7@adiguna-Suryo  ~/belajar/rust/learn-rust/oddman2  master ?  ./oddman
5
```

3. Pancake Sort
   Mengurutkan array dari yang terkecil hingga terbesar
   Array Input: [5, 4, 3, 2, 1]
   Kode Program:

```
#[inline]
fn reverse_until_index<T>(v: &mut [T], index: usize) {
    v[..index+1].reverse();
}
#[inline]
fn index_of_max<T>(v: &[T], index: usize) -> usize where T:
PartialOrd {
    let mut max = &v[0];
    let mut max_index = 0;
    for i in 0..(index+1) {
        if v[i] > *max {
            max = &v[i];
            max_index = i;
        }
    }
    max_index
}
#[inline]
pub fn pancake_sort<T>(v: &mut [T]) where T: PartialOrd {
    if v.len() == 0 {
        return;
    }
    let mut index = v.len() - 1;

    while index > 0 {
        let max_index = index_of_max(&v, index);
        if max_index != index {
```

```
            reverse_until_index(&mut *v, max_index);
            reverse_until_index(&mut *v, index);
        }
        index -= 1;
    }
}

fn main(){
    let mut v = vec![5, 4, 3, 2, 1];
    pancake_sort(&mut v);
    print!("{:?}", v);
}
```

Output:

```
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/pancake2  master ?  rustc pancake.rs
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/pancake2  master ?  ./pancake
[1, 2, 3, 4, 5]
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/pancake2  master ?
```

4. Reverse Words

   Membalikkan kata - kata dari suatu kalimat dengan urutan kata terakhir ke kata pertama

   Kode Program:

```
const SPACE: u8 = b' ';
#[inline]
pub fn ascii_reverse_words(s: &mut String) {
    let len = s.len();
    if len == 0 {
        return;
    }

    if !s.is_ascii() {
        panic!("Unexpected non-ASCII string: \"{}\"", s);
    }

    unsafe {
        let ref mut bytes = s.as_mut_vec();  // Unsafe.
        bytes.reverse();

        let mut left = 0;
        while left < len {
            if bytes[left] == SPACE {
```

```
                left += 1;
            } else {
                let mut right = left;
                while right < len && bytes[right] != SPACE {
                    right += 1;
                }
                bytes[left..right].reverse();
                left = right;
            }
        }
    }
}

fn main(){
    let mut s = "Hello World".to_string();
    ascii_reverse_words(&mut s);
    print!("{}", s);

}
```

Output:

```
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/reversewords2  master ? rustc reverseword.rs
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/reversewords2  master ? ./reverseword
World Hello
```

5. Targetsum

Mencari pasangan integer dari sebuah array yang jika ditambahkan sesuai dengan angka yang diinginkan

Array Input: [1, 2, 5, 10]

Target Input: 11

Kode Program:

```
pub fn target_sum(search: &[i32], target: i32) -> Option<(i32, i32)> {
    if search.len() < 2 {
        return None;
    }
    let mut sorted_search = search.to_vec();
    sorted_search.sort();
    let mut lowest_index = 0;
    let mut highest_index = sorted_search.len() - 1;

    while lowest_index < highest_index {
```

```
        let low = sorted_search[lowest_index];
        let high = sorted_search[highest_index];

        if low + high == target {
            return Some((low, high));
        } else if low + high > target {
            highest_index -= 1;
        } else {
            lowest_index += 1;
        }
    }
    None
}

fn main(){
    print!("{:?}", target_sum(&vec![1, 2, 5, 10], 11));
}
```

Output:

6. Printmult
   Print table dengan ukuran n * n dengan element bernilai baris * column
   n Input: 13
   Kode program:

```
fn main() {
    for i in 1..14 {
        for j in 1..14 {
            print!("{:4}", i*j);
        }
        print!("\n");
    }
}
```

Output:

```
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/printmult2  master ?  rustc printmult.rs
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/printmult2  master ?  ./printmult
  1   2   3   4   5   6   7   8   9  10  11  12  13
  2   4   6   8  10  12  14  16  18  20  22  24  26
  3   6   9  12  15  18  21  24  27  30  33  36  39
  4   8  12  16  20  24  28  32  36  40  44  48  52
  5  10  15  20  25  30  35  40  45  50  55  60  65
  6  12  18  24  30  36  42  48  54  60  66  72  78
  7  14  21  28  35  42  49  56  63  70  77  84  91
  8  16  24  32  40  48  56  64  72  80  88  96 104
  9  18  27  36  45  54  63  72  81  90  99 108 117
 10  20  30  40  50  60  70  80  90 100 110 120 130
 11  22  33  44  55  66  77  88  99 110 121 132 143
 12  24  36  48  60  72  84  96 108 120 132 144 156
 13  26  39  52  65  78  91 104 117 130 143 156 169
```

7. Sumfile
   Menambahkan bilangan2 yang berada didalam file, numbers.txt
   Input bilangan dalam file:
   52638321787
   855779448
   41250908755
   71531002628
   4317243210
   82348396266
   76837242877
   26332391104
   86624438235
   28584150665

   Kode Program:

```rust
use std::fs::File;
use std::io::BufRead;
use std::io::BufReader;

fn sum_file(path: &str) -> i64 {
    let mut sum : i64 = 0;
    let file = match File::open(path) {
        Ok(f) => f,
        Err(e) => panic!("couldn't open {}: {}", path, e),
    };
    let reader = BufReader::new(file);

    for readline in reader.lines() {
        let line = match readline {
            Ok(readline) => readline,
```

```
            Err(e) => panic!("couldn't read from {}: {}", path,
e),
        };
        match line.trim().parse::<i64>() {
            Ok(v) => sum += v,
            Err(_) => panic!("invalid integer in {}: {}", path,
line),
        }
    }
    sum
}

fn main() {
    println!("Sum: {}", sum_file("numbers.txt"));
}
```

Output:

```
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/sumfile2  master ?  rustc sumfile.rs
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/sumfile2  master ?  ./sumfile
Sum: 471319874975
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/sumfile2  master ?
```

8. Substring

   Menentukan apakah substring yang diberikan, ada pada sebuah kata
   Input kata : “abcdef”
   Input substring: “def”

   Kode Program:

```
pub fn has_substring(search: &str, find: &str) -> bool {
    if search.is_empty() {
        return find.is_empty();
    }

    let search_bytes = search.as_bytes();
    let find_bytes = find.as_bytes();
    let max_search_index = search_bytes.len() - find_bytes.len();

    for i in 0..(max_search_index + 1) {
        let mut found = true;
        for j in 0..find_bytes.len() {
            if search_bytes[i + j] != find_bytes[j] {
                found = false;
```

```
                break;
            }
        }
        if found {
            return true;
        }
    }
    false
}

fn main(){
    print!("{}",has_substring("abcdef", "def"));
}
```

Output:

```
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/substring2  master ?  rustc substring.rs
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/substring2  master ?  ./substring
true
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/substring2  master ?
```

9. Minstack
   Implementasi mencari nilai minimum stack pada rust

   Kode program:

```
pub struct MinStack<T> {
    stack: Vec<T>,
    min_stack: Vec<usize>
}

impl<T : Ord> MinStack<T> {
    #[inline]
    pub fn new() -> MinStack<T> {
        MinStack {
            stack: Vec::new(),
            min_stack: Vec::new()
        }
    }
    #[inline]
    pub fn push(&mut self, value: T) {
        let idx = self.stack.len();

        match self.min_stack.last() {
            Some(&min) if value <= self.stack[min] =>
```

```
                self.min_stack.push(idx),
            None =>
                self.min_stack.push(idx),
            _ => { }
        }

        self.stack.push(value);
    }
    #[inline]
    pub fn pop(&mut self) -> Option<T> {
        let value = self.stack.pop();

        match self.min_stack.last() {
            Some(&min) if min == self.stack.len() => {
                self.min_stack.pop();
            }
            _ => { }
        }

        value
    }
    #[inline]
    pub fn min(&self) -> Option<&T> {
        self.min_stack.last().map(|&n| &self.stack[n])
    }
}


fn main(){
    let mut stack : MinStack<i32> = MinStack::new();
    stack.push(3);
    stack.push(2);
    stack.push(1);

    print!("{:?}", stack.min());
}
```

Output:

```
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/minstack2 master ? rustc minstack.rs
adiguna7@adiguna-Suryo ~/belajar/rust/learn-rust/minstack2 master ? ./minstack
Some(1)
```

10. Rectangle

Mencari tahu, apakah suatu persegi panjang bisa muat dalam persegi panjang yang lain

Kode program:

```
#[derive(Debug)]
struct Rectangle {
    width: u32,
    height: u32,
}

impl Rectangle {
    fn square(size: u32) -> Rectangle {
        Rectangle { width: size, height: size }
    }
    fn area(&self) -> u32 {
        self.width * self.height
    }
    fn can_hold(&self, r2: &Rectangle) -> bool {
        self.width > r2.width && self.height > r2.height
    }
}

fn main() {
    let rect1 = Rectangle { width: 30, height: 50 };
    let rect2 = Rectangle { width: 10, height: 40 };
    let rect3 = Rectangle { width: 60, height: 45 };

    println!("The area of rect 1 is {}", rect1.area());

    println!("Can rect1 hold rect2? {}", rect1.can_hold(&rect2));
    println!("Can rect1 hold rect3? {}", rect1.can_hold(&rect3));
}
```

Output: