

# Match - exercise

## 1. Traffic Light Controller

&lt;/&gt; Rust

```
1 enum TrafficLight {
2     Red,
3     Yellow,
4     Green,
5 }
6
7 // Implement a function that returns the time each
  light should remain active
8 fn light_duration(light: TrafficLight) -> u8 {
9     todo!("Return duration in seconds: Red=30,
  Yellow=5, Green=45")
10 }
11
```

## 2. File Extension Analyzer

&lt;/&gt; Rust

```
1 // Write a function that returns the file type based
  on extension
2 fn file_type(extension: &str) -> &'static str {
3     todo!("Match extensions: 'txt'='Text',
  'rs'='Rust', 'jpg'='Image', etc.")
4 }
5
```

## 3. Payment Processor

&lt;/&gt; Rust

```
1 enum PaymentMethod {
2     CreditCard(String, u8, u16), // number, exp_month,
  exp_year
3     PayPal(String), // email
4     Crypto(String), // wallet address

```

```
5     Cash,  
6 }  
7  
8 // Implement a function that processes payment and  
   returns a confirmation message  
9 fn process_payment(method: PaymentMethod, amount: f64)  
   -> String {  
10     todo!("Return different messages for each payment  
        method")  
11 }  
12
```

## 4. Robot Direction Controller

&lt;/&gt; Rust

```
1 enum Direction {  
2     Up,  
3     Down,  
4     Left,  
5     Right,  
6 }  
7  
8 struct Position {  
9     x: i32,  
10    y: i32,  
11 }  
12  
13 // Implement movement logic that updates position  
   based on direction  
14 fn move_robot(current: Position, dir: Direction) ->  
   Position {  
15     todo!("Return new position after moving one unit  
        in given direction")  
16 }  
17
```

## 5. User Role Permissions

&lt;/&gt;

```
1 enum UserRole {  
2     Guest,  
3     User,  
4     Moderator,  
5     Admin,  
6 }  
7  
8 // Implement a function that returns permissions for
```

```
each role
9 fn get_permissions(role: UserRole) -> Vec<&'static
  str> {
10     todo!("Return different permission sets for each
      role")
11 }
12
```