## P1

```
// This program computes GCD of two numbers
// read from the command line using the iterative
Euclid algorithm.
fn main() -> void {
    let x: int = read();
    let y: int = read();

    while x != 0 {
        let old_x = x;
        x = y % x;
        y = old_x;
    }

    match y {
        y < 0 => print(-y),
        _ => print(y)
    }
}
```

## P2

```
// This program reads an integer from the command
// line and checks if it's 0, a digit or something
else.
fn main() -> void {
    let x: int = read();

    let result: str = match x {
        0 => "It's zero",
        1..10 => "It's a digit",
        _ => "It's something else"
    };

    print(result);
}
```

## P3

```
// This program retrieves the
// longest string from strings sent as arguments.
fn main(args: []str) -> void {
    let max_length: int = 0;
    let max_length_str: str = "";

    let aux: str;
    let length: int;
    for arg in args {
        aux = arg;
        length = 0;
        while aux != "" {
            length++;
            aux = aux[:-1];
        }
        if length > max_length {
            max_length = length;
            max_length_str = arg;
        }
    }

    print(max_length_str);
}
```

## Perr

```
fn main() -> void {
    // variables shouldn't begin with digits and
shouldn't
    // contain special characters
    // (":" is used for type adnotation and ")" is
used for grouping)
    let 321a:)x: int;

    // char literal too long
    if 321:)x > 0 {print('greater than zero'); }
    else { print("smaller than zero"); }
}
```