p1.fsz

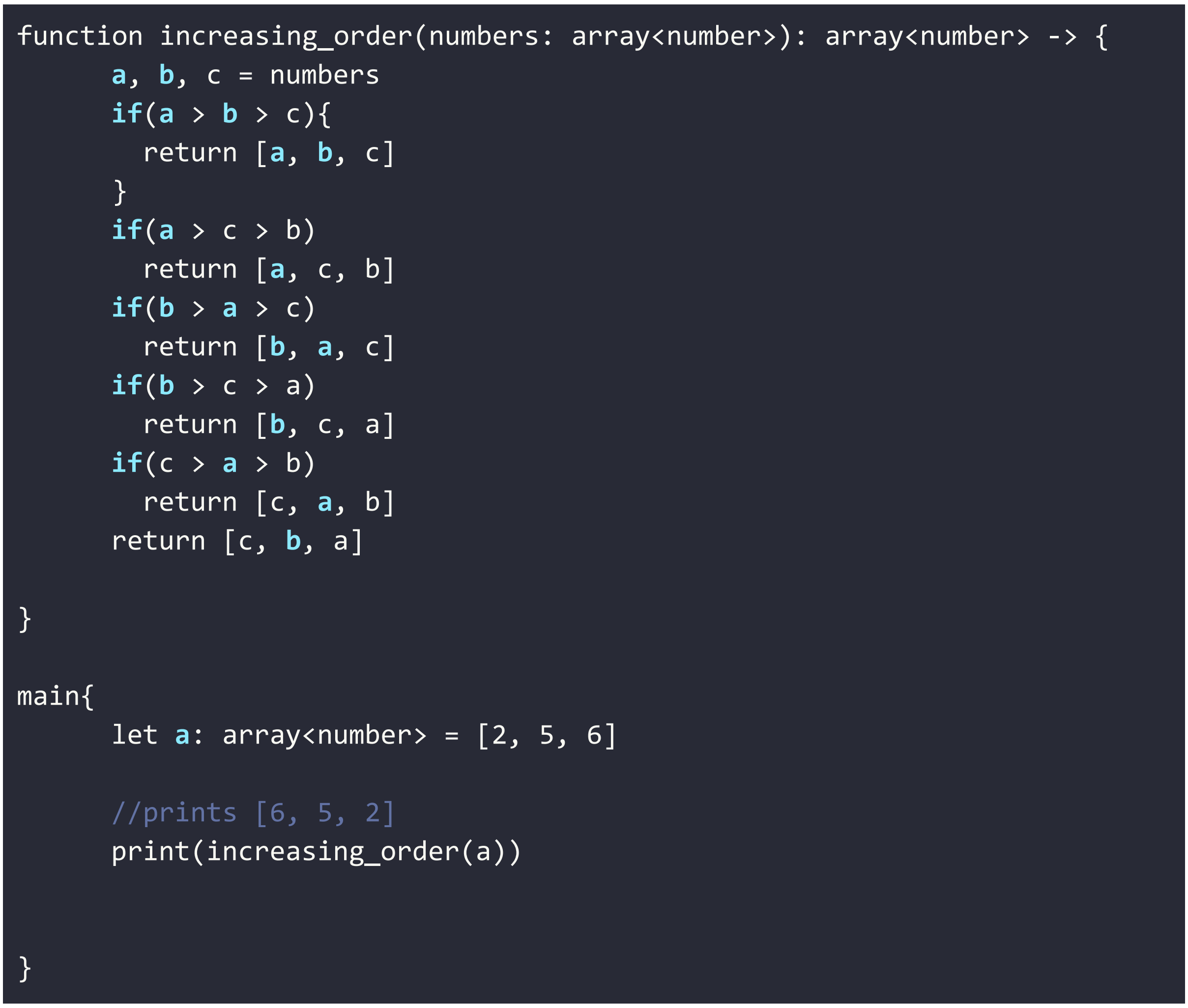
```
/This is a line comment

// This
      is
      a
      block
//  Comment


/define a number and an address and a function
let a: integer = 3;
let b: address = null;
let c: function = (a: integer) => &a


//prints the address of a
print(c(b))
```

p2.fsz



```
function increasing_order(numbers: array<number>): array<number> -> {
      a, b, c = numbers
      if(a > b > c){
        return [a, b, c]
      }
      if(a > c > b)
        return [a, c, b]
      if(b > a > c)
        return [b, a, c]
      if(b > c > a)
        return [b, c, a]
      if(c > a > b)
        return [c, a, b]
      return [c, b, a]

}

main{
      let a: array<number> = [2, 5, 6]

      //prints [6, 5, 2]
      print(increasing_order(a))


}
```

p3.fsz

```
//packages
from io import console



//
Lambda with return value
      func -> whatever
Lambda without return value
      func |> whatever

//

//variable number of arguments
function handle_response(...args: number): void {
      //loops over elements, and a switch statement
      when args{
      401 |> print("Unauthorized")
      404 |> print("Not found")
      else |> print("Error")
      }
}
main{
      number_of_codes: int = console()
      //type deduced from constant        //every value is immutable so
no i++
      error_codes: array[number]
      for(let i = 0; i < number_of_codes; i = i + 1){
      error_codes[i] = console()
      }
      handle_response(error_codes)
}
```

p1err.fsz

```
let a : int = 3 // good
a : int = 3 // bad, variables defined with let keyword



 /syntax error: whatever     is not defined
function(a: whatever) : void |> print(whatever) //lexical error
“function(a:” is not a recognized token

function asd(b: whatever): void |> print(b) /good
```