True or false? (unless otherwise specified)

1. In the Julia REPL, the key `;` enables to access *package mode*, from where software libraries can be installed and uninstalled.

2. In Julia, the first piece of code below (left) produces an error, while the second (right) plots the sine function (assuming that package `Plots` is already installed).

```
import Pkg
Pkg.add("Plots")
plot(cos + sin)
```

```
using Plots
plot(sin)
```

3. Complete the following code so that `f(n)` returns the n-th element f_n of the Fibonacci sequence $(f_1, f_2, f_3, \dots)$. Recall that, by definition, $f_1 = f_2 = 1$ and $f_n = f_{n-1} + f_{n-2}$ for $n \geqslant 3$.

```
f(n) = n in (1, 2) ? 1 :
```

4. In Julia, the following boolean expression is `true`:

```
sin([1.0, 2.0, 3.0]) == [sin(1.0), sin(2.0), sin(3.0)]
```

5. In the following code, the value of `a` is 15. (You may find it useful to read the documentation for the `sum` function to answer this question.)

```
a = sum(n -> n^2 - 1, [1, 2, 3])
```

6. In Julia, the command `[1, 2, 3] .* [1, 2, 3]` returns the array `[1, 4, 9]`.

7. In the following code, the variable `a` is an array (more precisely, a `Vector`) containing $\sqrt{34}$ and $\sqrt{160}$.

```
hypotenuse(x, y) = sqrt(x^2 + y^2)
a = hypotenuse.([3, 5], [4, 12])
```

8. The following piece of code prints `2`.

```
f(x::Float64) = 1
f(x::Int) = 2.0
@show f(f(1))
```

9. In a Jupyter notebook, the output displayed below a cell comes from the last expression evaluated in that cell. If that expression returns a value (like a number, a string or a plot object), Jupyter automatically displays it.

10. The following program prints the value 15.

```
result = 0
for i in 1:5
    result = result + i
end
println(result)
```

11. In Julia, arrays are indexed starting from `1`, not `0`.

12. In Julia, `=` is the equality operator, while `==` is the assignment operator.

13. In Julia, a function can be defined using the syntax `f(x) = x^2`

14. **Bonus.** The following code produces an assertion error

```
v = [[1, 2, 3], [2, 3, 4], [4, 5, 6]]
sum.(v) == [6, 9, 15]
```

Answer key

Questions 1–13: 1 point for a correct answer, 0 otherwise.

1. **False.** The key `]` gives access to *package mode*; the key `;` gives access to *shell mode*.

2. **True.** With `import Pkg` and `using Plots`, only the module `Pkg` is brought into scope by the first line, while `plot` is available after `using Plots`. Hence the first snippet raises an `UndefVarError` (and `plot(cos + sin)` would fail even with `plot` loaded), while the second plots the sine function.

3. The missing code is `f(n - 1) + f(n - 2)`:

   ```
   f(n) = n in (1, 2) ? 1 : f(n - 1) + f(n - 2)
   ```

4. **True.** `sin` broadcast over an array applies `sin` elementwise, so both sides produce the same vector.

5. **False.** The sum is $(1-1)+(4-1)+(9-1) = 0+3+8 = 11$, not 15.

6. **True.** `.*` multiplies the arrays elementwise, giving `[1*1, 2*2, 3*3] = [1, 4, 9]`.

7. **False.** Broadcasting gives `[sqrt(3^2 + 4^2), sqrt(5^2 + 12^2)] = [sqrt(25), sqrt(169)] = [5, 13]`.

8. **False.** `f(1)` dispatches to the `Int` method and returns `2.0`; `f(2.0)` then dispatches to the `Float64` method and returns `1`. So the program prints `1`, not `2`.

9. **True.** Jupyter displays the value of the last expression evaluated in a cell, if that expression returns a value.

10. **True.** The loop adds the integers $1+2+3+4+5 = 15$ and prints 15.

11. **True.** Julia arrays are 1-indexed.

12. **False.** `==` is the equality operator and `=` is the assignment operator; the statement has them reversed.

13. **True.** A short form function definition `f(x) = x^2` is valid Julia.

14. **False.** `sum.(v)` broadcasts the sum over the outer vector and equals `[6, 9, 15]`, so the comparison in the snippet is `true` and no assertion error is produced.