

Task v1

For numbers from 1 to 20:

- Print that number
- If number divides by 3, print `pa` instead
- If number divides by 5, print `pow` instead
- If number divides by both 3 and 5, print `papow` instead

Put spaces in between of the elements (but not before first or after the last one).

Expected output:

```
1 2 pa 4 pow pa 7 8 pa pow 11 pa 13 14 papow 16 17 pa 19 pow
```

Task v2

Maintain the code from task v1, it's still used in the system.

Besides that, for numbers from 1 to 15:

- Print that number
- If number is even (divides by 2), print `hatee` instead
- If number divides by 7, print `ho` instead
- If number is even and divides by 7, print `hateeho` instead

Put dashes in between of the elements (but not before first or after the last one).

Expected output:

```
1-hatee-3-hatee-5-hatee-ho-hatee-9-hatee-11-hatee-13-hateeho-15
```

As task v1 should be maintained, full expected output would be something like this:

```
Task v1:
1 2 pa 4 pow pa 7 8 pa pow 11 pa 13 14 papow 16 17 pa 19 pow
Task v2:
1-hatee-3-hatee-5-hatee-ho-hatee-9-hatee-11-hatee-13-hateeho-15
```

It would be really great if there would be no copy-and-paste (you *can* use copy-and-paste, just avoid duplicated code in the end result).

Refactoring, if needed

Before going on with task v3, let's discuss the code and/or implement some additional functionality.

Task v3

Before or while performing this task, consider refactoring your code into OOP.

Maintain the code from task v1 and v2, as it's still used in the system.

For numbers from 1 to 10:

- Print that number

- If number is one of the numbers 1, 4, 9, print `joff` instead
- If number is larger than 5, print `tchoff` instead
- If number is one of the numbers 1, 4, 9 and is larger than 5, print `jofftchoff` instead

Put dashes in between of the elements (but not before first or after the last one).

Expected output:

```
joff-2-3-joff-5-tchoff-tchoff-tchoff-jofftchoff-tchoff
```

As task v1 and v2 should be maintained, full expected output would be something like this:

```
Task v1:
1 2 pa 4 pow pa 7 8 pa pow 11 pa 13 14 papow 16 17 pa 19 pow
Task v2:
1-hatee-3-hatee-5-hatee-ho-hatee-9-hatee-11-hatee-13-hateeho-15
Task v3:
joff-2-3-joff-5-tchoff-tchoff-tchoff-jofftchoff-tchoff
```

It would be really great if there would be no copy-and-paste (you *can* use copy-and-paste, just avoid duplicated code in the end result).

Parameters and logic can be changed or added - it's best if your code would be easily maintainable, extensible and optionally testable.

For example, more conditions can be added for new tasks (less than 3, between 5 and 11, is primary etc.), this should be easily added to the code.