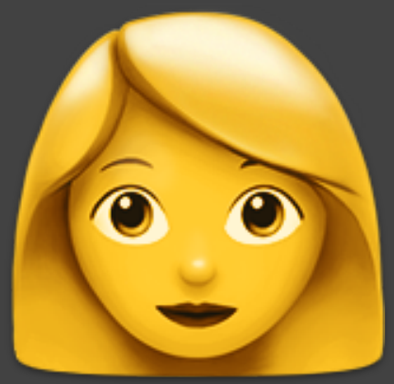
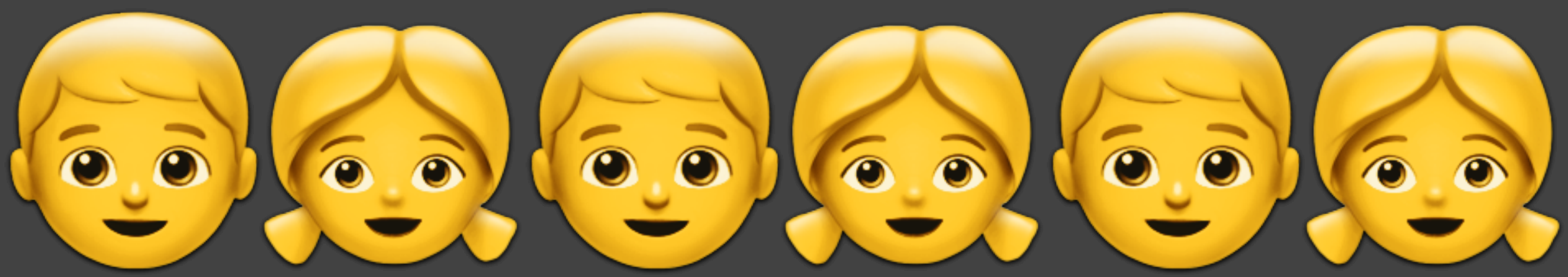
A Low Overhead Automated Service for Teaching Programming

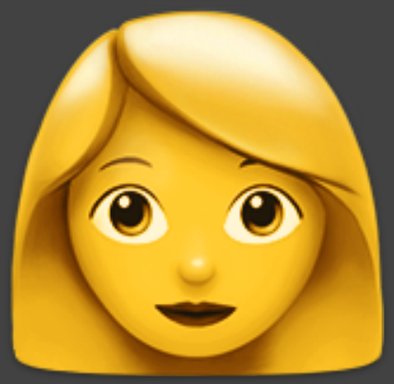
Matija Lokar & Matija Pretnar

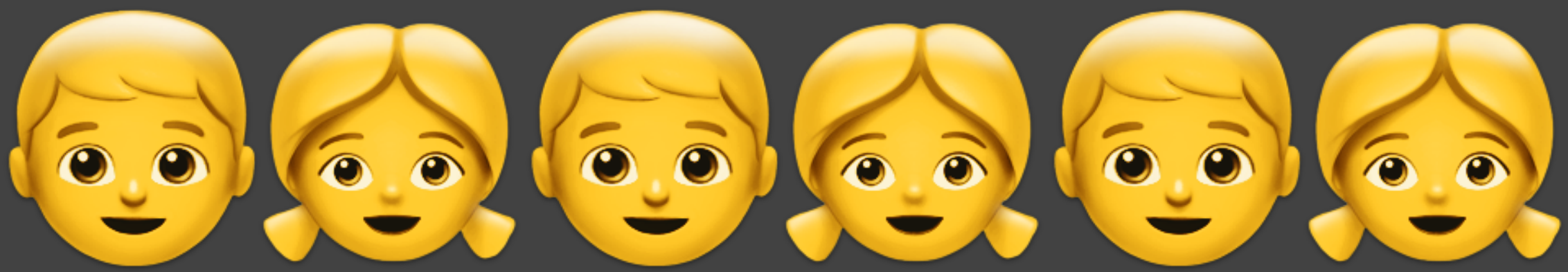
Faculty of Mathematics and Physics

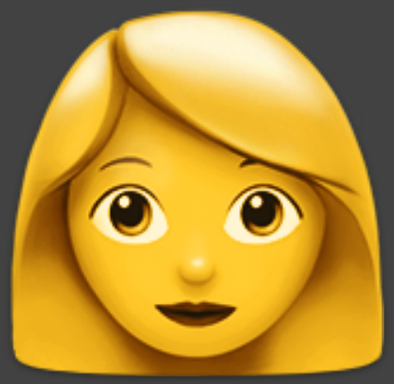
University of Ljubljana

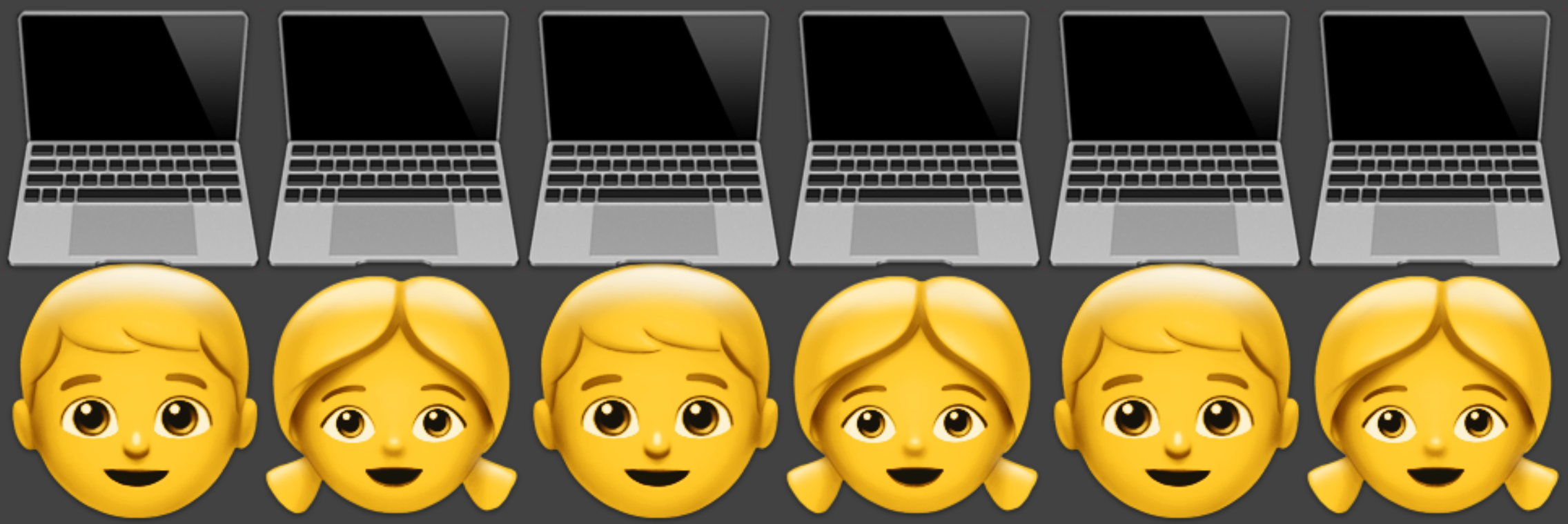


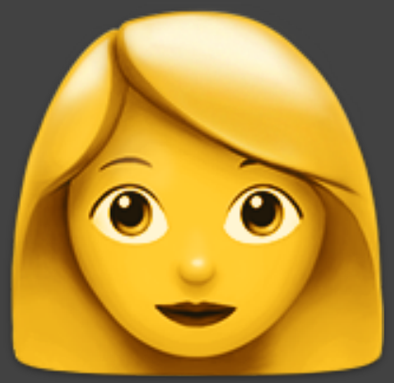


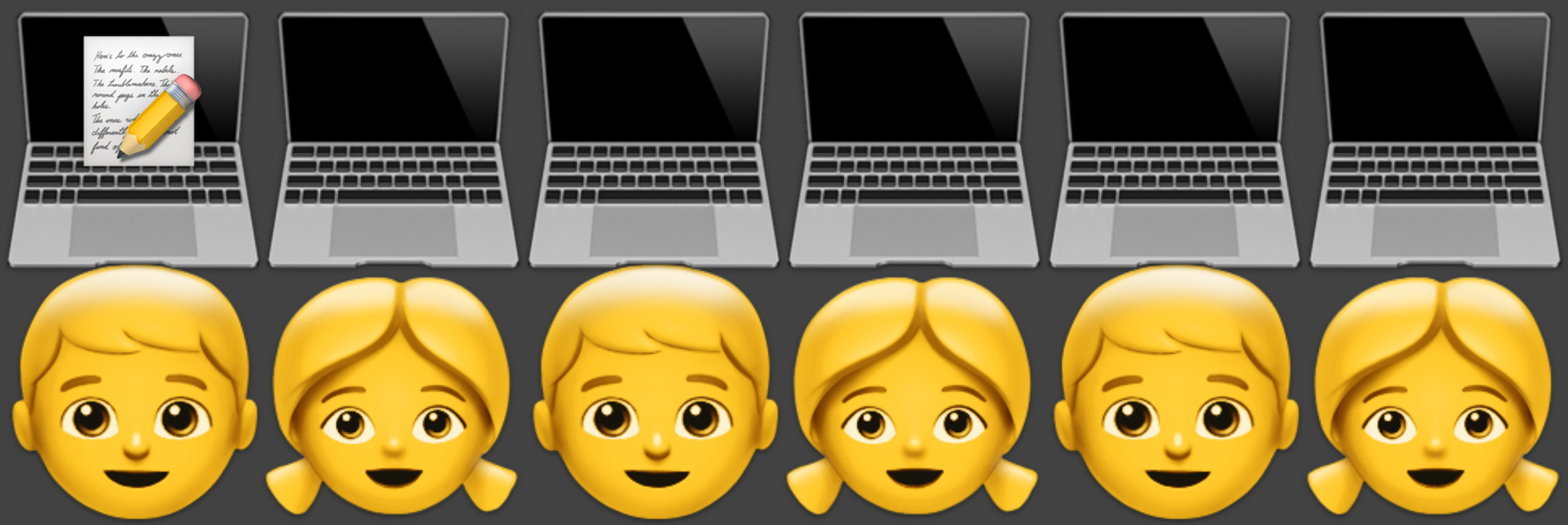


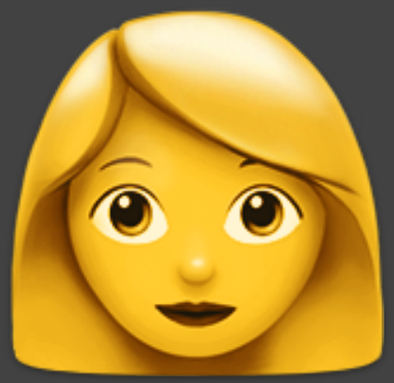


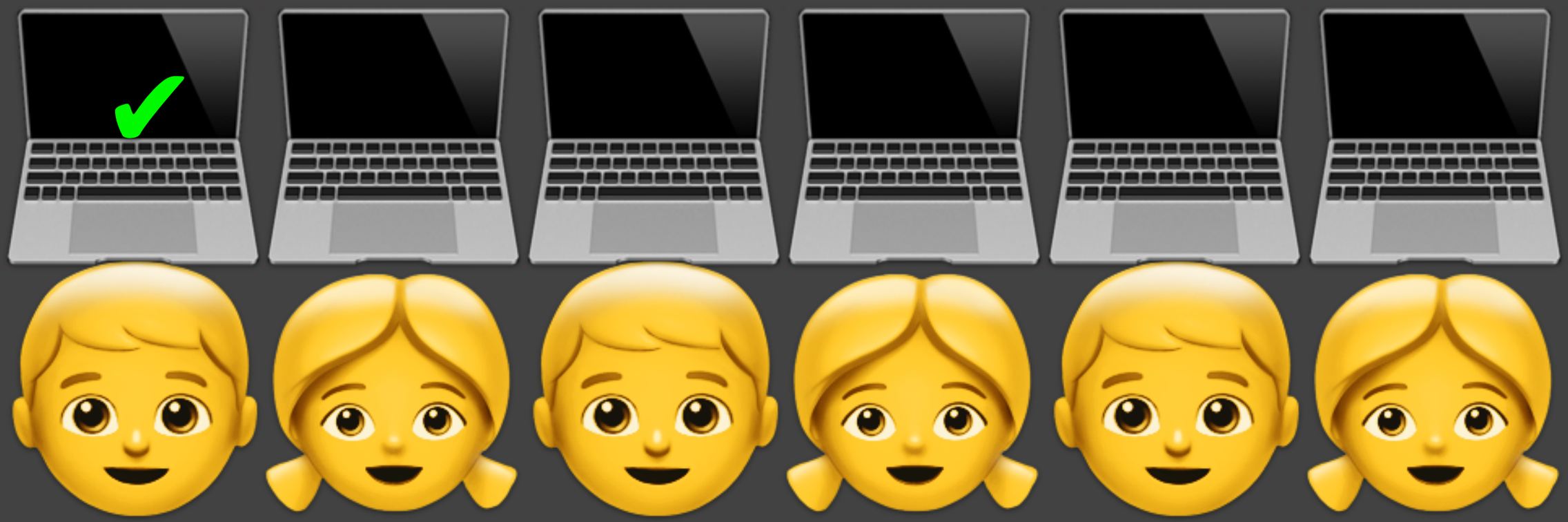


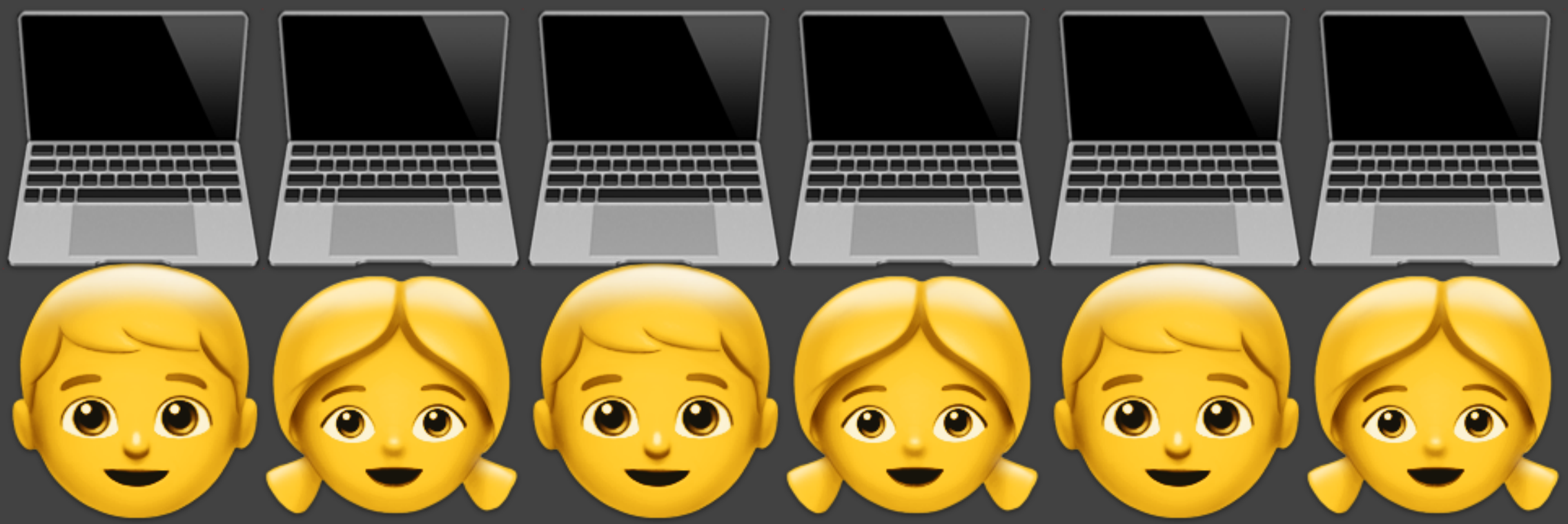


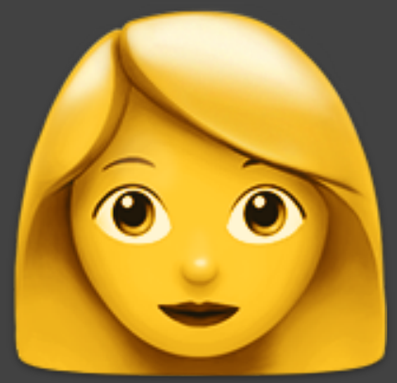


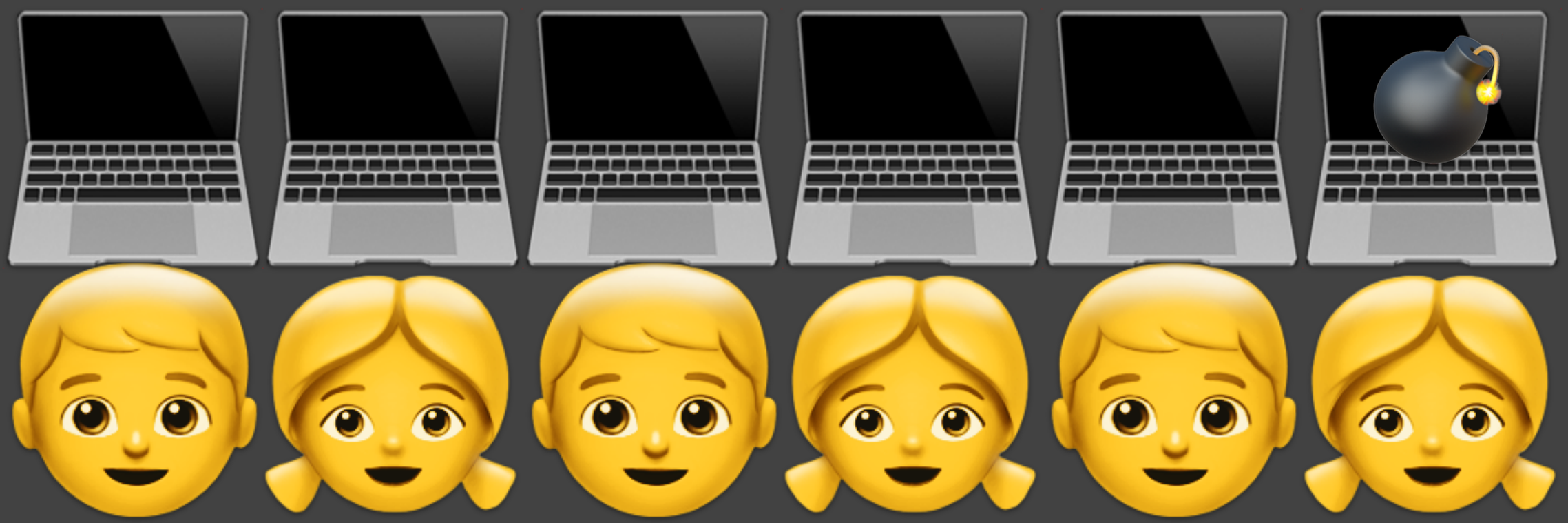


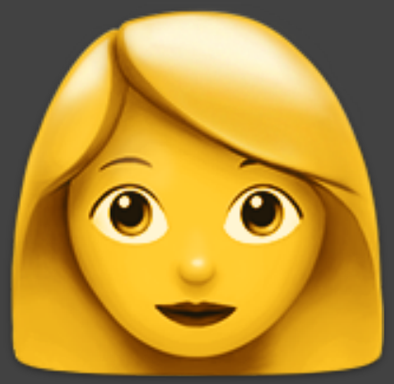


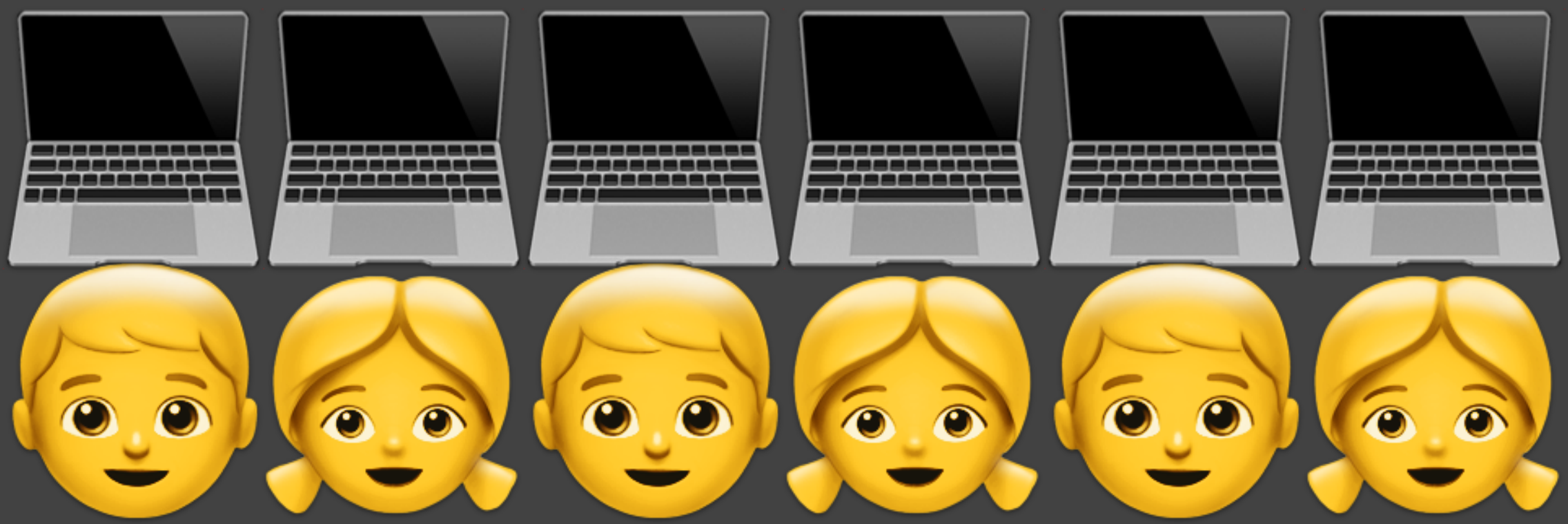


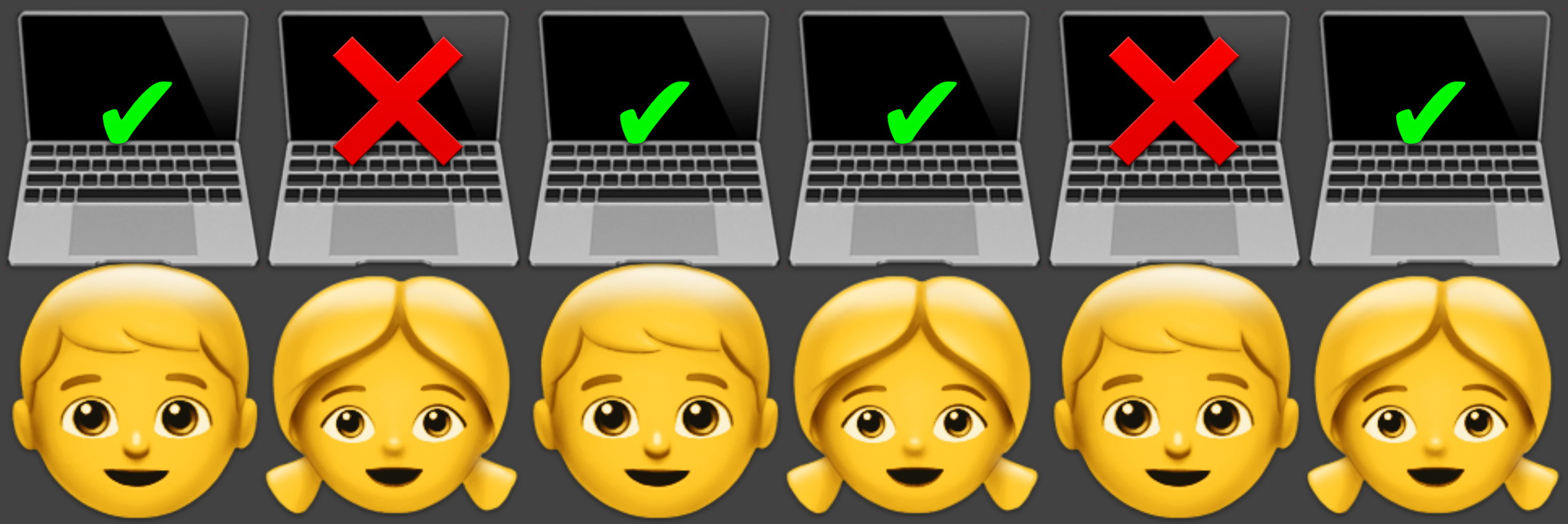


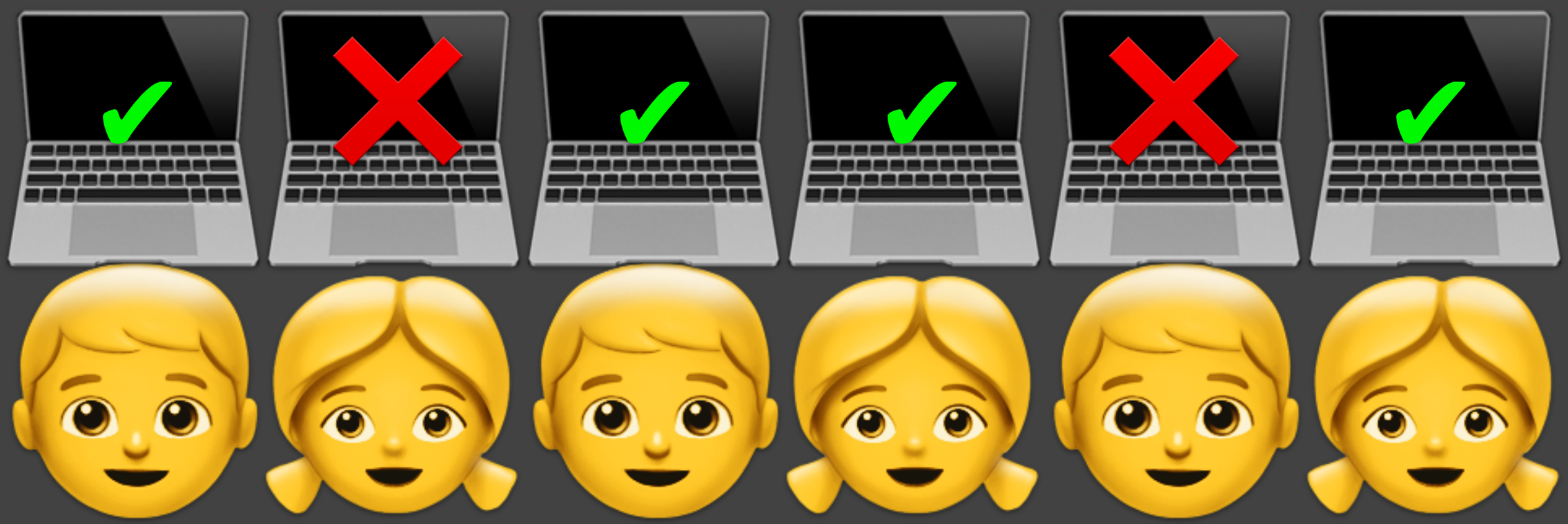












Projekt Tomo

v sodelovanju z Andrejem Bauerjem

http://tomo.fmf.uni-lj.si/problem_set/64/ — Tomo — Uvod v programiranje — Funkcije

Google

Tomo

Matija Pretnar

UVOD V PROGRAMIRANJE

- Delovno okolje 85%
- Izpis na zaslon 100%
- Operacije 33%
- Funkcije 0%
- Logične operacije 90%
- Pogojni stavek 40%

Funkcije

Reši sklop

Preproste funkcije

V Pythonu definiramo funkcijo takole:

```
def ime_funkcije(x,y,...,z):
    ukaz
    ukaz
    ...
    ukaz
```

Rezultat vrnemo z ukazom `return`.

Podnaloga 1 NEREŠENA

Sestavi funkcijo `linearna(a,b)`, ki sprejme števili `a` in `b` ter vrne rešitev enačbe $ax + b = 0$.

Podnaloga 2 NEREŠENA

Sestavi funkcijo `ploscina(n,a)`, ki sprejme števili `n` in `a` ter vrne ploščino pravilnega n-kotnika s stranico a.

Podnaloga 3 NEREŠENA

Sestavite funkcijo `prestopno(leto)`, ki vrne `True`, če je leto `leto` prestopno, sicer pa vrne `False`.

Razdalje med točkami

Podnaloga 1 NEREŠENA

Sestavite funkcijo `ravninskaRazdalja(x1, y1, x2, y2)`, ki vrne razdaljo med točkama (`x1` , `y1`) in (`x2` , `y2`).

```
>>> ravninskaRazdalja(1, 2, 3, 4)
2.82842712475
```

Podnaloga 2 NEREŠENA

Sestavite funkcijo `polarnaRazdalja(r1, fi1, r2, fi2)`, ki vrne razdaljo med točkama (`r1` , `fi1`) in (`r2` , `fi2`) v ravnini, pri čemer so koordinate v polarnem zapisu, koti pa so izraženi v stopinjah.

```
>>> polarnaRazdalja(1, 30, 4, 90)
3.60555127546
```

Python Shell

```
Python 3.2.2 (default, Jan 20 2012, 06:45:22)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2336.1.00)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
```

Ln: 4 Col: 4

01-preproste-funkcije.py – /Users/matija/Downloads/funkcije/01-preproste-funkcije.py

```
##########################################################################@@#
# Preproste funkcije
#
# V Pythonu definiramo funkcijo takole:
#
# def ime_funkcije(x,y,...,z):
#     ukaz
#     ukaz
#     ...
#     ukaz
#
# Rezultat vrnemo z ukazom return.
##########################################################################@@#


##########################################################################@000536#
# 1) Sestavi funkcijo linearna(a, b), ki sprejme števili a in b ter
# vrne rešitev enačbe $a x + b = 0$.
##########################################################################000536@#



##########################################################################@000537#
# 2) Sestavi funkcijo ploscina(n, a), ki sprejme števili n in a ter vrne
# ploščino pravilnega $n$-kotnika s stranico $a$.
##########################################################################000537@#



##########################################################################@000538#
# 3) Sestavite funkcijo prestopno(leto), ki vrne True, če je leto leto
# prestopno, sicer pa vrne False.
##########################################################################000538@#
```

Ln: 1 Col: 0

Ln: 4 Col: 4

01-preproste-funkcije.py – /Users/matija/Downloads/funkcije/01-preproste-funkcije.py

```
##########################################################################@@#
# Preproste funkcije
#
# V Pythonu definiramo funkcijo takole:
#
# def ime_funkcije(x,y,...,z):
#     ukaz
#     ukaz
#     ...
#     ukaz
#
# Rezultat vrnemo z ukazom return.
##########################################################################@@#


##########################################################################@000536#
# 1) Sestavi funkcijo linearna(a, b), ki sprejme števili a in b ter
# vrne rešitev enačbe $a x + b = 0$.
##########################################################################000536@#



##########################################################################@000537#
# 2) Sestavi funkcijo ploscina(n, a), ki sprejme števili n in a ter vrne
```

Preproste funkcije

V Pythonu definiramo funkcijo takole:

```
def ime_funkcije(x,y,...,z):
    ukaz
    ukaz
    ...
    ukaz
```

Rezultat vrnemo z ukazom `return`.

Podnaloga 1 NEREŠENA

Sestavi funkcijo `linearna(a,b)`, ki sprejme števili `a` in `b` ter vrne rešitev enačbe $ax + b = 0$.

Podnaloga 2 NEREŠENA

Sestavi funkcijo `ploscina(n,a)`, ki sprejme števili `n` in `a` ter vrne ploščino pravilnega n-kotnika s stranico a.

Podnaloga 3 NEREŠENA

Sestavite funkcijo `prestopno(leto)`, ki vrne `True`, če je leto `leto` prestopno, sicer pa vrne `False`.

01-preproste-funkcije.py – /Users/matija/Downloads/funkcije/01-preproste-funkcije.py

```
##########################################################################@@#
# Preproste funkcije
#
# V Pythonu definiramo funkcijo takole:
#
# def ime_funkcije(x,y,...,z):
#     ukaz
#     ukaz
#     ...
#     ukaz
#
# Rezultat vrnemo z ukazom return.
##########################################################################@@#


##########################################################################@000536#
# 1) Sestavi funkcijo linearna(a, b), ki sprejme števili a in b ter
# vrne rešitev enačbe $a x + b = 0$.
##########################################################################000536@#



##########################################################################@000537#
# 2) Sestavi funkcijo ploscina(n, a), ki sprejme števili n in a ter vrne
# ploščino pravilnega $n$-kotnika s stranico $a$.
##########################################################################000537@#



##########################################################################@000538#
# 3) Sestavite funkcijo prestopno(leto), ki vrne True, če je leto leto
# prestopno, sicer pa vrne False.
##########################################################################000538@#
```

Ln: 1 Col: 0

Ln: 4 Col: 4

```
##########################################################################@@#
# Preproste funkcije
#
# V Pythonu definiramo funkcijo takole:
#
# def ime_funkcije(x,y,...,z):
#     ukaz
#     ukaz
#     ...
#     ukaz
#
# Rezultat vrnemo z ukazom return.
##########################################################################@@#


##########################################################################@000536#
# 1) Sestavi funkcijo linearna(a, b), ki sprejme števili a in b ter
# vrne rešitev enačbe $a x + b = 0$.
##########################################################################000536@#
def linearna(a, b):
    return -b / a

##########################################################################@000537#
# 2) Sestavi funkcijo ploscina(n, a), ki sprejme števili n in a ter vrne
# ploščino pravilnega $n$-kotnika s stranico $a$.
##########################################################################000537@#



##########################################################################@000538#
# 3) Sestavite funkcijo prestopno(leto), ki vrne True, če je leto leto
# prestopno, sicer pa vrne False.
##########################################################################000538@#
```

01-preproste-funkcije.py - /Users/matija/Downloads/funkcije/01-preproste-funkcije.py

Python Shell

```
Python 3.2.2 (default, Jan 20 2012, 06:45:22)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2336.1.00)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
```

Ln: 4 Col: 4

01-preproste-funkcije.py - /Users/matija/Downloads/funkcije/01-preproste-funkcije.py

Python Shell

```
Python 3.2.2 (default, Jan 20 2012, 06:45:22)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2336.1.00)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ========================== RESTART ==========================
>>>
Shranjujem rešitve na strežnik... Rešitve so shranjene.
Podnaloga 1 je pravilno rešena.
Podnaloga 2 je brez rešitve.
Podnaloga 3 je brez rešitve.
>>>
```

Ln: 4 Col: 4

Funkcije

Reši sklop

Preproste funkcije

V Pythonu definiramo funkcijo takole:

```
def ime_funkcije(x,y,...,z):
    ukaz
    ukaz
    ...
    ukaz
```

Rezultat vrnemo z ukazom `return`.

Podnaloga 1 REŠENA
Sestavi funkcijo `linearna(a,b)`, ki sprejme števili `a` in `b` ter vrne rešitev enačbe $ax + b = 0$.

Podnaloga 2 NEREŠENA
Sestavi funkcijo `ploscina(n,a)`, ki sprejme števili `n` in `a` ter vrne ploščino pravilnega n-kotnika s stranico a.

Podnaloga 3 NEREŠENA
Sestavite funkcijo `prestopno(leto)`, ki vrne `True`, če je leto `leto` prestopno, sicer pa vrne `False`.

Razdalje med točkami

Podnaloga 1 NEREŠENA
Sestavite funkcijo `ravninskaRazdalja(x1, y1, x2, y2)`, ki vrne razdaljo med točkama (`x1` , `y1`) in (`x2` , `y2`).

```
>>> ravninskaRazdalja(1, 2, 3, 4)
2.82842712475
```

Podnaloga 2 NEREŠENA
Sestavite funkcijo `polarnaRazdalja(r1, fi1, r2, fi2)`, ki vrne razdaljo med točkama (`r1` , `fi1`) in (`r2` , `fi2`) v ravnini, pri čemer so koordinate v polarnem zapisu, koti pa so izraženi v stopinjah.

```
>>> polarnaRazdalja(1, 30, 4, 90)
3.60555127546
```

01-preproste-funkcije.py – /Users/matija/Downloads/funkcije/01-preproste-funkcije.py

```
##########################################################################@@#
# Preproste funkcije
#
# V Pythonu definiramo funkcijo takole:
#
# def ime_funkcije(x,y,...,z):
#     ukaz
#     ukaz
#     ...
#     ukaz
#
# Rezultat vrnemo z ukazom return.
##########################################################################@@#


##########################################################################@000536#
# 1) Sestavi funkcijo linearna(a, b), ki sprejme števili a in b ter
# vrne rešitev enačbe $a x + b = 0$.
##########################################################################000536@#
def linearna(a, b):
    return -b / a

##########################################################################@000537#
# 2) Sestavi funkcijo ploscina(n, a), ki sprejme števili n in a ter vrne
# ploščino pravilnega $n$-kotnika s stranico $a$.
##########################################################################000537@#
def ploscina(n, a):
    return n * a ** 2

##########################################################################@000538#
# 3) Sestavite funkcijo prestopno(leto), ki vrne True, če je leto leto
# prestopno, sicer pa vrne False.
##########################################################################000538@#
```

Ln: 28 Col: 21

Ln: 10 Col: 4

01-preproste-funkcije.py - /Users/matija/Downloads/funkcije/01-preproste-funkcije.py

Python Shell

```
Python 3.2.2 (default, Jan 20 2012, 06:45:22)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2336.1.00)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ========================== RESTART ==========================
>>>
Shranjujem rešitve na strežnik... Rešitve so shranjene.
Podnaloga 1 je pravilno rešena.
Podnaloga 2 je brez rešitve.
Podnaloga 3 je brez rešitve.
>>>
```

Ln: 10 Col: 4

01-preproste-funkcije.py - /Users/matija/Downloads/funkcije/01-preproste-funkcije.py

Python Shell

```
Python 3.2.2 (default, Jan 20 2012, 06:45:22)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2336.1.00)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> ========================== RESTART ==========================
>>>
Shranjujem rešitve na strežnik... Rešitve so shranjene.
Podnaloga 1 je pravilno rešena.
Podnaloga 2 je brez rešitve.
Podnaloga 3 je brez rešitve.
>>> ========================== RESTART ==========================
>>>
Shranjujem rešitve na strežnik... Rešitve so shranjene.
Podnaloga 1 je pravilno rešena.
Podnaloga 2 ni prestala vseh testov:
- Izraz ploscina(3,1) vrne 3 namesto 0.4330127018922193 (numerična napaka).
- Izraz ploscina(4,3) vrne 36 namesto 9 (numerična napaka).
- Izraz ploscina(5,2) vrne 20 namesto 6.881909602355868 (numerična napaka).
Podnaloga 3 je brez rešitve.
>>>
```

Ln: 10 Col: 4

Kako vse skupaj deluje?

Prednosti takega pristopa

takojšen odziv

varno izvajanje

poceni izvajanje

podpora več jezikom

Vprašanja?

Hvala!