

## Zad 1

Między liczbami parzystymi i nieparzystymi trwa wielka wojna. Wiele numerów straciło już życie na tej wojnie i Twoim zadaniem jest to zakończyć. Musisz określić, która grupa jest większa: parzysta czy nieparzysta. Większa grupa wygrywa.

Utwórz funkcję, która pobiera tablicę liczb całkowitych, osobno sumuje liczby parzyste i nieparzyste, a następnie zwraca różnicę między sumą liczb parzystych i nieparzystych.

Przykłady:

```
warOfNumbers([2, 8, 7, 5]) → 2
// 2 + 8 = 10
// 7 + 5 = 12
// 12 jest większe niż 10, zatem 12 - 10 = 2

warOfNumbers([12, 90, 75]) → 27

warOfNumbers([5, 9, 45, 6, 2, 7, 34, 8, 6, 90, 5, 243]) → 168
```

## Zad 2

Utwórz funkcję, która filtruje tablicę, aby uwzględnić liczby, które mają tylko określoną liczbę cyfr.

Przykłady:

```
filterDigitLength([88, 232, 4, 9721, 555], 3) → [232, 555]
// zwrócona powinna zostać tablica której liczby mają 3 cyfry każda.

filterDigitLength([2, 7, 8, 9, 1012], 1) → [2, 7, 8, 9]
// zwrócona powinna zostać tablica której liczby mają po jednej cyfrze
.

filterDigitLength([32, 88, 74, 91, 300, 4050], 1) → []
// brak liczb jednocyfrowych => zwrócona zostaje pusta tablica.

filterDigitLength([5, 6, 8, 9], 1) → [5, 6, 8, 9]
// wszystkie liczby mają jedną cyfrę => zwrócona zostaje oryginalna tablica
```

Uwagi:

- Jeśli nie istnieją żadne liczby o określonej długości cyfry, zwróć pustą tablicę.
- Jeśli wszystkie liczby w tablicy mają określoną długość cyfr, zwróć oryginalną tablicę.
- Zwrócona tablica podrzędna powinna mieć taką samą względną kolejność jak oryginalna tablica.

## Zad 3

Napisz funkcję, która zwraca tablicę z rozszerzeniami plików.

Przykłady:

```
getExtension(["code.html", "code.css"])
→ ["html", "css"]

getExtension(["project1.jpg", "project1.pdf", "project1.mp3"])
→ ["jpg", "pdf", "mp3"]

getExtension(["ruby.rb", "cplusplus.cpp", "python.py", "javascript.js"])
→ ["rb", "cpp", "py", "js"]
```