

Challenge 1o1

Tasya Amanda Adinegara

1. Function

```
circuit_power(230, 10) → 2300
```

```
circuit_power(110, 3) → 330
```

```
circuit_power(480, 20) → 9600
```

2. Buat fungsi yang menggunakan dua string sebagai argumen dan mengembalikan True atau False bergantung pada apakah jumlah total karakter di string pertama sama dengan jumlah total karakter di string kedua.

```
comp("AB", "CD") → True
```

```
comp("ABC", "DE") → False
```

3. Buat fungsi yang mengambil string (akan menjadi nama depan dan belakang seseorang) dan mengembalikan string dengan nama depan dan belakang ditukar.

```
name_shuffle("Donald Trump") → "Trump Donald"
```

```
name_shuffle("Rosie O'Donnell") → "O'Donnell Rosie"
```

```
name_shuffle("Seymour Butts") → "Butts Seymour"
```

4. Buat fungsi yang mengambil daftar angka antara 1 dan 10 (tidak termasuk satu angka) dan mengembalikan angka yang hilang.

```
missing_num([1, 2, 3, 4, 6, 7, 8, 9, 10]) → 5
```

```
missing_num([7, 2, 3, 6, 5, 9, 1, 4, 8]) → 10
```

```
missing_num([10, 5, 1, 2, 4, 6, 8, 3, 9]) → 7
```

5. Buat fungsi yang mengambil dua string angka dan mengembalikan jumlahnya sebagai string.

```
add("111", "111") → "222"
```

```
add("10", "80") → "90"
```

```
add("", "20") → "Invalid Operation"
```

6. Buat fungsi yang membangun kata dari huruf acak yang ada di list pertama. Gunakan list kedua untuk menetapkan setiap posisi huruf dalam daftar pertama. Kembalikan string dari huruf yang tidak diacak (yang menyusun kata).

```
word_builder(["g", "e", "o"], [1, 0, 2]) → "ego"
```

```
word_builder(["e", "t", "s", "t"], [3, 0, 2, 1]) → "test"
```

7. Tulis fungsi yang mengubah setiap huruf menjadi huruf berikutnya:

"a" menjadi "b"

"b" menjadi "c"

"d" menjadi "e"

dan seterusnya ...

```
move("hello") → "ifmmp"
```

```
move("bye") → "czf"
```

```
move("welcome") → "xfmdpnf"
```

Tidak akan ada z dalam pengujian.

8. Tulis fungsi yang mengembalikan True jika dua list, saat digabungkan, membentuk urutan yang berurutan. Urutan berurutan adalah urutan tanpa celah di bilangan bulat, mis. 1, 2, 3, 4, 5 adalah urutan yang berurutan, tetapi 1, 2, 4, 5 false.

```
consecutive_combo([7, 4, 5, 1], [2, 3, 6]) → True
```

```
consecutive_combo([1, 4, 6, 5], [2, 7, 8, 9]) → False
```

```
consecutive_combo([1, 4, 5, 6], [2, 3, 7, 8, 10]) → False
```

```
consecutive_combo([44, 46], [45]) → True
```

9. Diberikan integer, buat fungsi yang mengembalikan bilangan prima berikutnya. Jika bilangan prima, kembalikan bilangan itu sendiri.

```
next_prime(12) → 13
```

```
next_prime(24) → 29
```

```
next_prime(11) → 11
```

```
# 11 is a prime, so we return the number itself.
```

10. Buat fungsi yang mengambil usia dan mengembalikan usia dalam hitungan hari.

```
calc_age(65) → 23725
```

```
calc_age(0) → 0
```

```
calc_age(20) → 7300
```

11. Dalam tantangan ini, seorang petani meminta Anda untuk memberi tahu dia berapa banyak kaki yang dapat dihitung di antara semua hewannya. Petani membiakkan tiga spesies:

ayam = 2 kaki

sapi = 4 kaki

babi = 4 kaki

Peternak telah menghitung hewannya dan dia memberi Anda subtotal untuk setiap spesies. Anda harus mengimplementasikan fungsi yang mengembalikan jumlah total kaki semua hewan.

*Ingatlah bahwa peternak ingin mengetahui jumlah total kaki dan bukan jumlah total hewan.

```
animals(2, 3, 5) → 36
```

```
animals(1, 2, 3) → 22
```

```
animals(5, 2, 8) → 50
```

12. Buat fungsi untuk menguji apakah sebuah string adalah pin yang valid atau tidak.

Pin yang valid memiliki:

Tepat 4 atau 6 karakter.

Hanya karakter numerik (0-9).

Tidak ada spasi.

```
valid("1234") → True
```

```
valid("45135") → False
```

```
valid("89abc1") → False
```

```
valid("900876") → True
```

```
valid(" 4983") → False
```

13. Buat fungsi yang mengambil string dan mengembalikan jumlah (hitungan) vokal yang terkandung di dalamnya.

```
count_vowels("Celebration") → 5
```

```
count_vowels("Palm") → 1
```

```
count_vowels("Prediction") → 4
```

14. Buat fungsi yang menggunakan bilangan bulat n dan mengembalikan faktorial faktorial. Lihat contoh di bawah untuk pemahaman yang lebih baik:

```
fact_of_fact(4) → 288  
# 4! * 3! * 2! * 1! = 288
```

```
fact_of_fact(5) → 34560
```

```
fact_of_fact(6) → 24883200
```

Terimakasih