Homework01_Guide

Homework01_Guide

1. A Plus Abs B

- The implementation of the function is simple.
- When `b` < 0 , the function returns `a - b`.
-
- When `b` > 0, the function returns `a + b`.

> ⚠ This code return f(a, b) in the end, so `f` is not a variable but a function.

```
from operator import add, sub


def a_plus_abs_b(a, b):
    if b < 0:
        f = sub
    else:
        f = add
    return f(a, b)
```

2. Two of Three

1. Implementation 1

- Add all the squares of the three numbers, and subtract the maximum square.

```
def two_of_three(i, j, k):
    return i ** 2 + j ** 2 + k ** 2 - max(i, j, k) ** 2
```

2. Implementation 2

- Choose two numbers and add their squares, and compare the three results.

```
def two_of_three(i, j, k):
    return min(i * i + j * j, i * i + k * k, j * j + k * k)
```

3. Largest Factor

- From `n - 1` to 1, find the largest factor of `n`.

```
def largest_factor(n):
    factor = n - 1
    while factor > 0:
        if n % factor == 0:
            return factor
        factor -= 1
```

4. Hailstone

- If n is even, divide it by 2.
- If n is odd, multiply it by 3 and add 1.
- Continue this process until n is 1.

⚠ If `n == 1` initially, then the sequence is one step long!

```
def hailstone(n):
    length = 1
    while n != 1:
        print(n)
        if n % 2 == 0:
            n = n // 2
        else:
            n = 3 * n + 1
```

```
        length = length + 1
    print(n)
    return length
```