1.

```
def find_equilibrium(arr):
    total = sum(arr)
    left_sum = 0
    for i, num in enumerate(arr):
        total -= num
        if left_sum == total:
            return i
        left_sum += num
    return -1

arr = list(map(int, input().split()))
print(find_equilibrium(arr))
```

2.

```
def find_flavors(cost, money):
    index_map = {}
    for i, price in enumerate(cost):
        if money - price in index_map:
            return index_map[money - price] + 1, i + 1
        index_map[price] = i
    return None

money = int(input())
cost = list(map(int, input().split()))
print(find_flavors(cost, money))
```

3.
```
def find_fractions(arr):
    n = len(arr)
    pos = sum(1 for x in arr if x > 0) / n
    neg = sum(1 for x in arr if x < 0) / n
    zero = sum(1 for x in arr if x == 0) / n
    print(f"{pos:.3f}\n{neg:.3f}\n{zero:.3f}")

arr = list(map(int, input().split()))
find_fractions(arr)
```

```
4. def find_items(words, char):
    return [word for word in words if word.startswith(char)]

words = input().split()
char = input()
print(find_items(words, char))
```

```
5. def remove_consecutive_duplicates(arr):
    return [arr[i] for i in range(len(arr)) if i == 0 or arr[i] != arr[i-1]]

arr = list(map(int, input().split()))
print(remove_consecutive_duplicates(arr))
```

```
6. def run_length_encode(arr):
    result = []
    count = 1
    for i in range(1, len(arr)):
        if arr[i] == arr[i-1]:
            count += 1
        else:
            result.append([arr[i-1], count] if count > 1 else arr[i-1])
            count = 1
    result.append([arr[-1], count] if count > 1 else arr[-1])
    return result

arr = list(map(int, input().split()))
print(run_length_encode(arr))
```

7. def sort_nested_list(lst):

```
    return sorted(lst, key=lambda x: x[-1])

lst = eval(input())
print(sort_nested_list(lst))
```

8. arr = [[['*' for _ in range(2)] for _ in range(4)] for _ in range(3)]

```
print(arr)
```

9. def add_matrices(A, B):

```
    return [[A[i][j] + B[i][j] for j in range(len(A[0]))] for i in range(len(A))]

A = eval(input())
B = eval(input())
print(add_matrices(A, B))
```

10. def max_sum_list(lst):

```
    return max(lst, key=sum)

lst = eval(input())
print(max_sum_list(lst))
```

11. def remove_duplicates(lst):

```
    return [list(x) for x in set(tuple(x) for x in lst)]

lst = eval(input())
print(remove_duplicates(lst))
```

12. def rotate_strings(strings, n):

```
    return [s[n:] + s[:n] for s in strings]
```

```
strings = input().split()
n = int(input())
print(rotate_strings(strings, n))
```

13.
```
def common_indices(lists):
    return [i for i in range(len(lists[0])) if all(lst[i] == lists[0][i] for lst in lists)]

lists = eval(input())
print(common_indices(lists))
```

14.
```
import datetime

def format_date(date_str):
    date_obj = datetime.datetime.strptime(date_str, "%m/%d/%Y")
    return date_obj.strftime("%B %d, %Y")

date_str = input()
print(format_date(date_str))
```

15.
```
print(input().title())
```

16.
```
def row_sums(matrix):
    return [sum(row) for row in matrix]

matrix = eval(input())
print(row_sums(matrix))
```