```python
import statistics as st
import doctest


def compute_budget(total_budget: float, citizen_votes: list) -> list:
    """
       A method that receives as input the amount of money in the coffers and
the citizens' votes,
        and calculates the budget using the generalized median algorithm with
linear ascending functions

       Parameters
       ----------
       total_budget:
           The total budget

       citizen_votes:
           List of the citizens' votes

       Examples
       --------
       >>> budget = compute_budget(100, [[100, 0, 0], [0, 0, 100]])
       >>> print(budget)
       [50.0, 0, 50.0]
       >>> budget = compute_budget(30, [[0, 0, 6, 0, 0, 6, 6, 6, 6],[0, 6, 0,
6, 6, 6, 6, 0, 0],[6, 0, 0, 6, 6, 0, 0, 6, 6]])
       >>> print(budget)
       [2.0, 2.0, 2.0, 4.0, 4.0, 4.0, 4.0, 4.0, 4.0]
       """
    return binary_search(0, 1, citizen_votes, total_budget)


def f_votes(size: int, t: float, c: float) -> float:
    const_votes = []
    for i in range(1, size):
        const_votes.append(c * min(1, i * t))
    return const_votes


def binary_search(low, high, citizen_votes, total_budget):
    if high >= low:
        mid = (high + low) / 2
        all_medians = medians(citizen_votes, f_votes(len(citizen_votes), mid,
total_budget))
        sum_of_medians = sum(all_medians)
        if sum_of_medians == total_budget:
            return all_medians

        elif sum_of_medians > total_budget:
            return binary_search(low, mid, citizen_votes, total_budget)

        else:
            return binary_search(mid, high, citizen_votes, total_budget)

    else:
        return -1
```

```
def medians(citizen_votes: list, const_votes: list):
    all_medians = []
    for idx in range(len(citizen_votes[0])):
        current_median = const_votes.copy()
        for current_vote in citizen_votes:
            current_median.append(current_vote[idx])
        all_medians.append(st.median(current_median))
    return all_medians


if __name__ == '__main__':
    (failures, tests) = doctest.testmod(report=True)
    print("{} failures, {} tests".format(failures, tests))
```