

1) Develop the programs in c, c++, java or python to solve the following problem statements

Activity 1:

Given an array, of integers, calculate and print the respective mean, median, and mode on separate lines. If your array contains more than one modal value, choose the numerically smallest one.

Input format:

1. The first line contains an integer, N, denoting the number of elements in the array.
2. The second line contains N space-separated integers describing the array's elements.

Output format:

Print lines of output in the following order:

1. Print the mean on a new line, to a scale of decimal place
2. Print the median on a new line, to a scale of decimal place
3. Print the mode on a new line; if more than one such value exists, print the numerically smallest one.

Constraints:

- N belongs to [10, 2500]
- X belongs to [0, 10000]

Solution:

```
import numpy as np
from scipy import stats
size = int(input())
numbers = list(map(int, input().split()))
print(np.mean(numbers))
print(np.median(numbers))
print(int(stats.mode(numbers)[0]))
```

- 2) Given an array, A, of N integers and an array, W, representing the respective weights of A's elements, calculate and print the weighted mean of A's elements. Your answer should be rounded to a scale of decimal place

Input format:

1. The first line contains an integer, N , denoting the number of elements in arrays A and W .
2. The second line contains N space-separated integers describing the respective elements of array A .
3. The third line contains N space-separated integers describing the respective elements of W array.

Output format:

- Print the weighted mean on a new line. Your answer should be rounded to a scale of 1 decimal place (i.e., format).

Constraints:

- $N=[5 \text{ to } 50]$
- $a_i=[0 \text{ to } 100]$
- $W_i=[0 \text{ to } 100]$

Solution:

```
N = map(int,input().split())
X = list(map(int, input().strip().split(' ')))
W = list(map(int, input().strip().split(' ')))
sum_X = sum([a*b for a,b in zip(X,W)])
print(round((sum_X/sum(W)),1))
```

3) Given an array, X , of n integers, calculate the respective first quartile ($Q1$), second quartile ($Q2$), and third quartile ($Q3$). It is guaranteed that $Q1, Q2, Q3$ are integers.

Input Format

- The first line contains an integer, n , denoting the number of elements in the array.
- The second line contains n space-separated integers describing the array's elements.

Output format:

Print lines of output in the following order:

1. The first line should be the value of Q1.
2. The second line should be the value of Q2.
3. The third line should be the value of Q3 .

Constraint:

1. $n=[5 \text{ to } 50]$
2. $x_i=[0 \text{ to } 100]$

Solution:

```
n = int(input())
x = sorted(list(map(int, input().split())))
from statistics import median
print(int(median(x[:n//2])))
print(int(median(x)))
print(int(median(x[(n+1)//2:])))
```

4) The interquartile range of an array is the difference between its first (Q1) and third (Q3) quartiles. Given an array, X, of n integers and an array, F, representing the respective frequencies of X's elements, construct a data set, S, where each x_i occurs at frequency f_i . Then calculate and print S's interquartile range, rounded to a scale of 1 decimal place.

Input format:

The first line contains an integer, n , denoting the number of elements in arrays X and F.

The second line contains n space-separated integers describing the respective elements of array X.

The third line contains n space-separated integers describing the respective elements of array F.

Output Format:

Print the interquartile range for the expanded data set on a new line. Round your answer to a scale of 1 decimal place.

Solution:

```
import statistics as st
n = int(input())
data = list(map(int, input().split()))
freq = list(map(int, input().split()))
```

```
s = []
for i in range(n):
    s += [data[i]] * freq[i]
N = sum(freq)
s.sort()
```

```
if n%2 != 0:
    q1 = st.median(s[:N//2])
    q3 = st.median(s[N//2+1:])
else:
    q1 = st.median(s[:N//2])
    q3 = st.median(s[N//2:])
```

```
ir = round(float(q3-q1), 1)
print(ir)
```
