Problem B: High-speed Signal Equalization

Introduction

This contest consists of two problems. This problem is about recovering a digital signal.

High-speed interface, as the key part of communication between chips, plays a crucial role in computing, network, wireless, terminal and other fields. Due to the non-ideal characteristics of analog devices and transmission channels in communication links, the received signals of high-speed interfaces are often accompanied by specific impairments. With the continuous evolution of communication rates, in addition to linear impairments, end-to-end communication signals also contain strong nonlinearity and noise. The increasingly complex damage components make the system analysis and algorithm design more complex.

A typical digital signal processing (DSP) assisted PAM4 communication system is shown in Fig. 1. A digital pattern sequence is coded in PAM4 pattern with the symbols $[-3, -1, 1, 3]$ to represent the 4 levels of a PAM4 signal. After TX DSP, the digital signal is converted to analog waveform using a digital to analog converter (DAC). After transmission by a high speed channel, the analog signal is then received by a receiver. The analog to digital converter (ADC) converts the analog signal to a digital signal. After equalizing and slicing by the RX DSP, the original digital signal from the transmitter is recovered at the receiver.

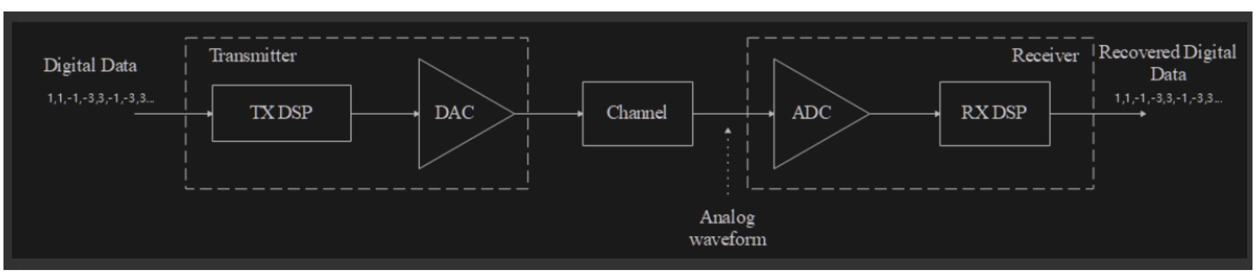

Figure 1: PAM4 DSP-assisted communication system

Challenges

The main challenges of this problem are as follows:

1. The damage in the channel is diverse and coupled, which stems from analog/optical device, passive link and active amplification, including noise, insertion loss, nonlinear-

ity, etc. The nonlinearity and noise are amplified in the band-limited channel when the transmission rate is increased.

2. The energy efficiency per bit is reduced with the rapid growth of transmission speed. The architecture faces a severe challenge of power.

In view of the above background, you should perform damage decomposition on the signal, and accurately model and analyze the signal. Under the given input data, the signal waveform is modeled and fitted, and the original data is restored from the analog waveform through the algorithm. In order to match the requirements of low power consumption, delay, and bit error rate in the actual communication system, this contest must be completed within the specified complexity and accuracy.

Task

You are given an analog waveform that comes from a PAM4 signal in a real channel. The waveform is represented as a sequence of double values. Each digital symbol is a unit interval (UI). The oversample rate (OSR) means the number of samples per UI for the analog waveform. It is always 4 in this task. This means that each UI has 4 samples in the analog waveform.

For example, if the waveform length is 4029972 samples, then the number of UIs is:

$$\frac{4029972}{4} = 1007493.$$

Your task is to use signal equalization to recover the transmitted symbols from the waveform, then slice them into one of the four possible PAM4 levels: $[3, 1, -1, -3]$.

Input

Your program should read input using standard input (stdin).

- The first line of input contains the integer N, the number of samples in the analog wave.

- The next line contains N space-separated real numbers $a_1, a_2, \ldots, a_N$, the contents of the analog wave. It is guaranteed that N is divisible by 4, and that $-1 \leq a_i \leq 1$ for each $1 \leq i \leq N$.

Note that there is a large amount of input data, so you should use an efficient input method to read the data. Documentation on how to do this in your language of choice can be found at:

`https://open.kattis.com/languages`

Case	N	Score Multiplier
Case 1	4029972	1
Case 2	4003932	1
Case 3	4128768	1
Case 4	4029972	1
Case 5	4029972	1
Case 6	4063108	1
Case 7	4063108	2
Case 8	4029972	2

Figure 2: Challenges and goals

Output

Your program should write output to standard output (stdout).

- First, you should write an integer L, the number of samples in your digital signal. This number must be equal to $N/4$, or you will receive the verdict `Wrong Answer`.
- Then, on the next line, print L space-separated integers, your reconstruction of the digital signal. Each number must be one of $\{3, 1, -1, -3\}$.

Scoring

Your solution will be tested and scored on 8 testcases, which are described above. To prevent overfitting, these testcases are kept hidden.

To receive a score, you must output a digital signal within the time limit. For each testcase, the bit error rate (BER) is computed as follows:

$$\text{BER} = \frac{\text{Number of incorrect bits}}{\text{Total transmitted bits}}$$

Each testcase is then assigned a score between 0 and 20, with more points awarded for lower BER. This score is then multiplied by the score multiplier for that testcase. The total score for the problem is the sum of the score for each testcase. If you make multiple submissions, the one with highest total score is the one counted on the scoreboard.

Sample of the inverse problem of canal estimation

Your program should read input using standard input (`stdin`).

The first line of input contains the integer N, the number of samples in `ana_waveform_1`.

The next line contains N space-separated real numbers $a_1, a_2, \ldots, a_N$, the contents of `ana_waveform_1`. It is guaranteed that $-1 \le a_i \le 1$ for each $1 \le i \le N$.

The next line contains the integer M, the number of samples in `dig_data_1`. The oversampling rate between the analog and digital data is 16, which means that $N = M \cdot 16$ always holds in this problem.

The next line contains M space-separated integers, the contents of `dig_data_1`. Each of these integers is one of $\{3, 1, -1, -3\}$.

The next line contains the integer K, the number of samples in `dig_data_2`.

The final line contains K space-separated integers, the contents of `dig_data_2`. Each of these integers is one of $\{3, 1, -1, -3\}$.

Your solution will be tested on 4 testcases, whose characteristics are described in the following table.

Sample Input 1

```
16

0.22061 -0.70137 -0.55763 0.35266 0.96795 0.84704

-0.48585 0.55557 -0.46686 -0.98207 0.13065 0.85421

0.19748 -0.71182 0.38312 0.69907
1
3
2
-1 3
```

Sample Output 1

```
32
-0.52561 0.82934 -0.92622 -0.00621 -0.83611

0.61852 0.23000 -0.51393 -0.83801 -0.50055 0.63153

0.84901 0.50833 -0.71813 0.51604 0.02155 0.93926
```

```
-0.49804 0.43322 -0.28385 0.56230 -0.21097 0.20067

0.12074 -0.62109 -0.90709 -0.73369 -0.01097

0.16630 0.01916 -0.64849 0.63309
```

1 Hints for the competition organizer

1.1 Data from our problem

- *ana_waveform.in* : the wavefrom data with length 1000.
- *sliced_data.ans* : the corresponding sliced data with length 250.
- This data is a part of the test case 1 of problem 2.

1.2 Questions and answers

Question : If my answer is exactly the negative of the standard answer, can I still get points? It seems that if the input is only the *ana_waveform*, we cannot determine the sign of the *digital_data*, right?

Answer : *Dig_data* and *ana_waveform* correspond to the same form and can distinguish the sign of *digital_data*.

Question : My i know what modulation is used to convert pam4 into analog before distortion ?

Answer : The $analog_waveform$ is converted from PAM4 digital data by a DACdigital to analog converter). https://en.wikipedia.org/wiki/Digital-to-analog$_converter$

Question : Do we have any info on the treatment of the wave in the Tx phase? Was some filter applied before sending it?

Answer : Yes, a 3tap FFE (pre1,main,post1) digital filter was used before the digital data send to the DAC.

Yes, a 3tap FFE (pre1,main,post1) digital filter was used before the digital data send to the DAC.

Question : can we get at least one real input sample for each problem?

Answer : For problem 1, the *ana_waveform_1* is the answer which means the recovered analog waveform from *dig_data_1* should be similar to *ana_waveform_1* . For problem 2, the problem1 can be used as a reference. The *dig_data_1* is the answer extract from *ana_waveform_1*. A little tips: The ratio between sample input length and sample output length should be attention.