This is a test for candidates who wish to participate in the training class organized by VCB-Studio. Finish as many problems as you can, and then do the following things:

(1) Pack your answers, result files, and necessary attachments into a **zip/rar/7z** file. Source files we provided and intermediate file in your encoding should not be packed in.

(2) Register a Baidu Net Disk account (pan.baidu.com), upload the zipped file and create a sharing link. Whether you like it or not, Baidu Net Disk has been the most effective way to share files within our team since day one. Other sharing methods will NOT be considered.

(3) Send the link via email to vcbs.training@gmail.com before **Beijing Time (UTC+8) Monday, 23 Jan 2023, 23:59:59**. Late submissions will NOT be considered.

(4) Prepare a QQ account. The follow-up training courses will be conducted in the QQ group.

You should independently complete the answers without any public discussion. Any form of plagiarism will NOT be tolerated.

This test has 5 questions. For question 2 and 3, you can choose ONE of them. Choosing both then we will pick one with higher points. The answers should be made in English.

Question1 (15pt)

Please describe yourself as who you are, where do you study, how do you come to know VCB-Studio and why are you interested in this project, etc. Please do not write more than 500 words, or approximately 1 page. (15pt)

Question2 (30pt)

Scanned pictures (or simply scans) are an important part of BDRips, which are often released as lossless PNG, TIFF format or lossy JPG format. Scans feature high resolution and large size. In the file **Q2.7z**, two sets of pictures have been provided for you. PNGs are the source scans, and WEBPs are transcoded from PNGs according to VCB-Studio Collation specifications. Your tasks are:

(1). Summarize the format conversion rules of scans in VCB-Studio Collation specifications. (6pt)

(2). Convert the sources to AVIF and JPEG-XL format, with sizes comparable to the WEBPs. (12pt)

(3). Comment on the quality, encoding speed, and compatibility of AVIF and JPEG-XL, and why/why not you may recommend us switching to the new format as the upgrade for WEBP in 2023. (12pt)

You are free to utilize existing tools, but you need to describe clearly where you find the tool and how to use it.

Question3 (30pt)

Recently 32-bit audio tracks have appeared in some of the latest Hi-Res music. Although now we would not see these annoying 32-bit tracks in the Blu-ray, we have to start working on them in advance. In the file **Q3.7z**, two 32-bit PCM files are provided for you. Your tasks are:

(1). Learn about 32-bit tracks and tell the difference between these two files. (6pt)

(2). Try to convert them to FLAC, ALAC, and WavPack losslessly. (15pt)

(3). Consider various aspects such as compression rate, encoding speed, and playback compatibility and select the format you recommend most for 32-bit audio. (9pt)

You are free to utilize existing tools, but you need to describe clearly where you find the tool and how to use it.

Question4 (35pt)

MSU publishes video encoder tests every year, with the latest one here: https://compression.ru/video/codec_comparison/2021/main_report.html.

For the first time last year, H.266 (VVC) encoders participated in the tests and they performed well in terms of encoding quality in the slow encoding (1 fps) test.

(1). Choose any of the H.266 (VVC) or AV1 encoders in the figure below, and then encode the source file *Q4 [E46686C4].m2ts* with no more than 2500 Kbps of video bitrate. You'd better use 10bit variants of these encoders, which facilitates the comparison later. In addition, you need to describe clearly where you found the

encoder and state the version and parameters you used. If you use H.266 (VVC) encoder, you will get additional 5pt. (10pt+5pt)

(2). We provide an AV1 video file *Q4_AV1 [41A7EDDA].mkv*, which was encoded via SVT-AV1 10bit encoder without any pre-processing. Comment on the picture quality compared to the source file. When you compare the picture quality, you may want to sample a few frames, attach some screenshots, and comment on the performance of dark scenes and moving scenes. (10pt)

(3). Now compare your own encoding to the given AV1 file in terms of picture quality, encoding speed, and playback compatibility. As a reference, we encoded the above AV1 file at 1.0 fps. (10pt)
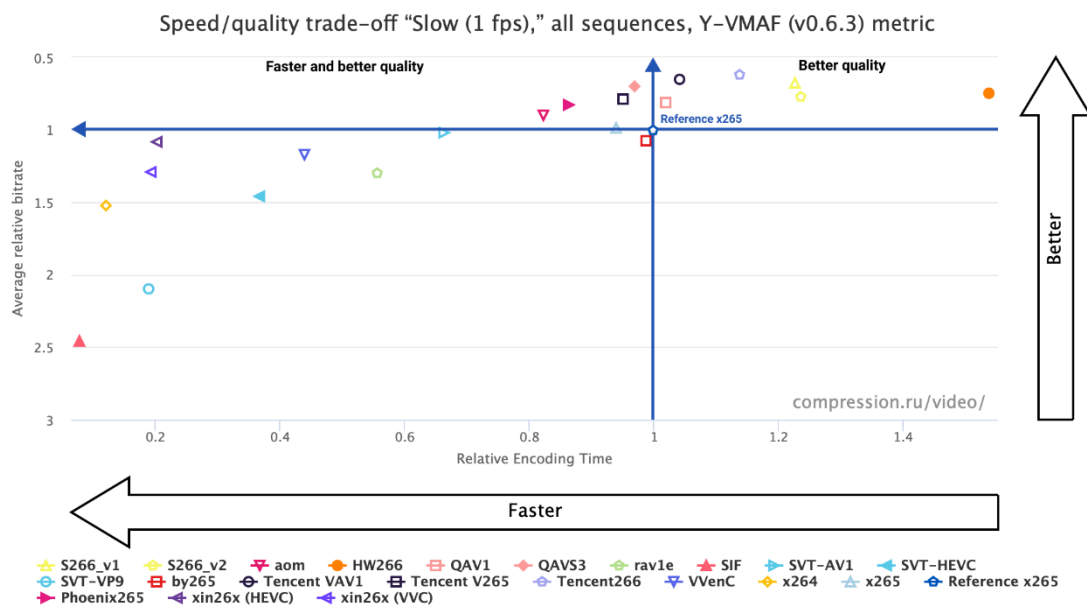
Question5 (20pt)

When we check an encoded file, we need to locate frames that have been encoded exceptionally awful. We use algorithms like PSNR to evaluate the similarity of each frame in the encoded file to the source file. The result is an array of scores, where

the i-th score is tied to the i-th frame. These scores are called raw scores. However, what we are concerned about is the standard score, which is the raw score minus a threshold. A frame with a standard score less than 0 is considered a bad frame. The tasks are:

(1) Find the worst frame, i.e. the one with the lowest standard score among the bad frames, and output its index. If there is more than one worst frame, output the first. If there are no bad frames, output -1. Frames with a standard score of exactly 0 are not considered as bad frames. (10pt)

Input:

2 lines. The first line is two integers that represent the number of frames N and the threshold value S. The second row is an array of integers $A[N]$, representing the raw score of each frame.

For all the data, $1 <= N <= 200000$, $0 < S < 100$, $0 <= A[i] <= 100$

Output:

An integer, the index of the worst frame. The index starts from 0. If there is more than one worst frame, output the first. If there are no bad frames, output -1.

Sample:

Input
10 30
42 31 44 23 21 26 31 41 50 72

Output
4

(2) Find a continuous sequence of frames that minimizes the sum of their standard scores and output this minimum value. Full scores will only be given if the time complexity of your algorithm is optimal. (10pt)

Input:

The same as (1).

Output:

An integer, the minimum sum value.

Sample:

Input
10 30
42 31 44 23 21 26 31 41 50 72
Output
-20

For each sub question, use C/C++/Java/Python/C# to write a console program. Read the input from the standard input and write it to standard output. Do NOT use libraries other than built-in ones (for example, no "import numpy as np"). Submit your source code.