DEVIALET

INGÉNIERIE ACOUSTIQUE DE FRANCE

Software Engineer – home exercise (recruitment)

Dear Candidate,

You have successfully passed the first step of our recruitment process.

We believe that one can't properly evaluate the programming skills by questions alone. The goal of this home exercise is to evaluate your coding skills in an environment as close as possible to the real one. You can take as much time as you want (although the exercise is tailored to take you no more than a few hours), use search engines, and, most importantly, organize it any way you like. The important things to remember:

- You should work on it in a way you'd work on a real task in a company.
- The provided solution should be complete, i.e. it must be in a state where you'd say to your manager: "the work on this project is 100% finished".
- You should be proud of your code and of using it as a showcase for your skills.

The description of the exercise is on the next page. You may find it quite imprecise, but this is done on purpose. We are eager to see how you interpret and complete such a task description.

During the live discussion in our offices, we'll start by analyzing and executing your code. You should be able to explain the structure of your code, its operation, and why you have structured it this way; advantages, disadvantages, and possible improvements.

Once the home exercise is covered, the discussion may be complemented by additional live exercises of smaller extents (on different subjects).

Please send the complete code to us by email or by posting to your GitHub account and providing us with a link (preferred).

Good coding!

Software development team

Imaginary player

The program should simulate basic operations of a playlist-capable music player.

The simulated “music” files (“tracks”) should include metadata such as the title, duration, codec…

The user interaction must include:

- Play, Pause, Next, Previous operations
- Random and repeat
- Managing the playlist:
    - Adding, removing tracks
    - Removing duplicates from the playlist
- Show track information
- Show playlist information

Actual music playback is not required. The user interface may be as simple as an interactive command console (see example below).

Language: C++

Platform/compiler: any

Libraries: STL only

Example (minimal, don’t hesitate to add other information which might be useful for the user):

```
$ iplayer
Imaginary player 1.0
>>> add_track 01_intro.music
Added 01_intro.music
>>> add_track 02_running.music
Added 02_running.music
>>> play
Playing 01_intro.music
>>> show_track
Track 1(2)
01_intro.music
Duration: 03:00
Title: Introduction
Codec: MP3
>>>
```