# Starfleet

## Interview

Staff 42 pedago@42.fr

*Summary: This document is an interview question for the Starfleet Piscine.*

# Contents

# Chapter I

# General rules

- The interview should last between `45 minutes`.
- Both the interviewer and the interviewed student must be present.
- The interviewed student should write his code using a `whiteboard`, with the language of her/his choice.
- At the end of the interview, the interviewer evaluates the student based on the provided criteria.

Read carefully the interview question and solutions, and make sure you `understand` them before the interview. You can't share this document with other students, as they might be interviewed on the same question. Giving them the answer would prevent them from having to solve an unknown question during an interview.

## I.1 During the interview

During the interview, we ask you to :

- Make sure the interviewed student `understands` the question.
- Give her/him any `clarification` on the subject that she/he might need.
- Let her/him come up with a solution before you guide her/him to the best solution given the constraints (time and space).
- Ask the student what is the `complexity` of her/his algorithm ? Can it be improved and how ?
- `Guide` her/him to the best solution without giving the answer. You may refer to the `hints` for that.
- You want to evaluate how the interviewed student thinks, so ask her/him to `explain everything` that she/he thinks or writes (there should be no silences).
- If you see a mistake in the code, wait untill the end and give her/him a chance to correct it by her/himself.
- Ask the student to show how the algorithm works on an `example`.
- Ask the student to explain how `limit cases` are handled.
- Bring out to the student any mistake she/he might have done.
- Give `feedback` on her/his performances after the interview.
- Be `fair` in your evaluation.

As always, stay mannerly, polite, respectful and constructive during the interview. If the interview is carried out smoothly, you will both benefit from it !

# Chapter II

# The game of beautiful numbers

## II.1 Interview question

Here is the description of the game :

Initially there is a number `n` written on the board.

Two players start playing the game turn by turn.

Each player has to replace the number `n` written on the board by n-2^k (for some k >= 0 such that 2^k < n).

Also the number n-2^k has to be as `beautiful` as n. The beauty of a number depends on the number of `1s` in its binary represntation.

A player loses the game when he can't select any k.

**`Given the initial number n, determine which player will win the game if both players play optimally.`**

`Note` : n > 0 and n <= 10^9

`Note` : It might be useful to write the rules of the game on the whiteboard.

### II.1.1 Hints

- Try to play the game with `n = 10` for instance.
- Write the binary represention of `n` turn by turn. Don't you see a pattern?

## II.2 Best solution

### II.2.1 Count the number of 0s

If `n = 10` at the begining, the game would go like this:

```
                           n = 10 [1010]
                       /                  \
                      /                    \
Player 1:   k = 0, n = 9 [1001]    k = 2, n = 6 [0110]
                     |                      |
Player 2:   k = 2, n = 5 [0101]    k = 0, n = 5 [0101]
                      \                    /
                       \                  /
Player 1:               k = 1, n = 3 [0011]
                                 |
Player 2:                    GAME OVER
```

n = 10 -> in binary representation

ROUND 1 - `Player 1` has 2 choices:

- k = 0: n-2^0 = 10 - 1 = 9
- k = 2: n-2^2 = 10 - 4 = 6

ROUND 2 - `Player 2` has only one choice in both cases:

- k = 2: n-2^2 = 9 - 4 = 5
- k = 0: n-2^0 = 6 - 1 = 5

ROUND 3 - `Player 1` has only one choice:

- k = 1: n-2^1 = 5 - 2 = 3

ROUND 4 - `Player 2` looses the game.

At each turn, the players select `k` such that the number of `1s` in `n` is preserved. In fact, they are just shifting one `1 bit` to the right. The game ends when no `1 bit` can be shifted to the right anymore.

The total number of rounds would be the number of `0s` before each `1 bit` set in n at the biginning of the game.

To know which player will win the game, we only need to check if the number of rounds (count of `0s` before each `1 bit` set in `n`) is even or odd.

```
O(m) time , O(1) space
where m is the length of the binary representation of n
```

code:

```
int selectWinner(int n) {
        int count = 0;

        while (n != 0) {
                if (n & 1 == 0)
                        count++;
                n >>= 1;
        }
        return ((count % 2) ? 2 : 1);
}
```