Environment

I used C++ version 13.0.0 to perform this program.
The below commands in a terminal can successfully run my program.

~ % g++ 110550091.cpp -o name
~ % ./name

Report

I first used a while loop to cin *n* indefinitely until EOF, then used a for loop inside the while loop to cin the *n* coins into a vector. The coin vector is then put into a function named *fake* along with its four pivots and vector size to find the fake coin.

The vector is separated into 3 large piles and possibly a leftovers pile if n is nondivisible by 3. The four pivots represents the beginning of each pile. Each weight of the 3 large pile is set to 0 initially, but they accumulate the weight of each coin in the respective pile eventually by the 3 for loops.

Later comes the comparisons. If all 3 piles are equal, then the only reasonable explanation is that the fake coin lies in the leftover pile. The leftover pile is either of size 1 or size 2 (given by the result of size module 3). If its size is 1, then return the last pivot (p4), else compare the 2 coins from a real coin (take a random coin from a pile that is all real, in this case, take the coin in the p1th index) and return the fake coin's index accordingly.

If one of the 3 piles are different from the other two, then it is considered if the size of that pile is below 3 or more. If it contains less than 3 coins, then it doesn't need to perform *fake* again. The same method as the leftover pile is treated will be applied here. On the other hand, if it contains 3 or more coins, then *fake* is performed again for the pile containing the fake coin in a recursive way.

The function 'fake' is performed recursively until the piles of coins containing the fake one are parted to contain less than 3 coins. Only one of the 3 coins piles are recursively performed. Moreover, in each run, the weight of each coins are summed up linearly. Therefore, the time complexity of this program would be:

$$T(n) = T(n/3) + f(n)$$

According to the master theorem,

$$a = 1 \quad b = 3 \quad f(n) = n$$
$$\log_b a = \log_3 1 = 0$$
$$n^0 < n^1$$

So *f(n)* will take dominance of the equation, meaning that the time complexity of this program is:

$$T(n) = \emptyset(f(n)) = \emptyset(n)$$

Overall, this project was quite fun. It took the same amount of time thinking about how to find the fake coin using only a scale as to actually turn my thoughts into a runnable program. Originally, I thought this project would take around 4 hours, but it took only around 2 hours to complete and an additional half an hour to write the report. The process of writing the program went surprisingly smooth without any setbacks. I really liked this homework!