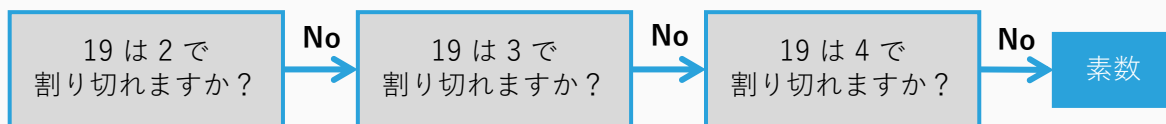


問題 3.1.1

自分の年齢 N が 2 歳以上の場合、以下のようにして判定すれば良いです。

- 2 以上 \sqrt{N} 以下の整数で割り切れない場合：素数
- そうでない場合：合成数

たとえば 19 歳の場合は、 $\sqrt{19} = 4.358 \dots$ より 2, 3, 4 で割れば良いですが、いずれでも割り切れないため素数です。



問題 3.1.2

まず、自然数 N を素因数分解したときに、 \sqrt{N} を超えるものは高々 1 つしかありません。これは背理法（→3.1.3項）を用いて次のように証明することができます。

自然数 N を素因数分解したときに \sqrt{N} を超えるものが 2 つ以上あると仮定する。すなわち、 \sqrt{N} を超える整数 A, B を用いて、以下のように素因数分解されると仮定する。

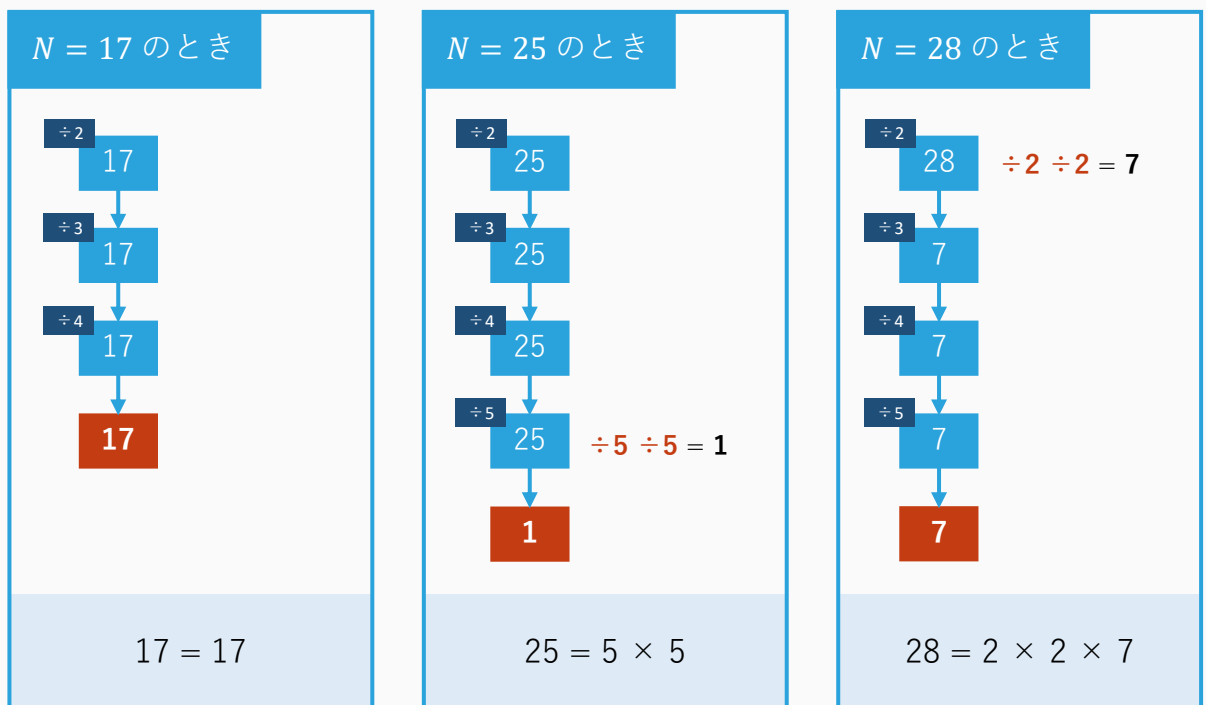
$$N = \bigcirc \times \bigcirc \times \dots \times \bigcirc \times A \times B$$

しかし、 $A \times B > N$ であるため矛盾する。よって \sqrt{N} を超えるものは 1 つ。

したがって、次のようなアルゴリズムで素因数分解することができます。

- N を 2 で割れるだけ割る。
- N を 3 で割れるだけ割る。
- 同じような操作を、 $4, 5, \dots, \lfloor \sqrt{N} \rfloor$ についても行う。
- 最後に残った N が 1 以外の場合、それを素因数に加える。

たとえば、 $N = 17, 25, 28$ の場合、下図のようにアルゴリズムが動作します。同じ数を二度以上割るケースもあることに注意してください。



C++ での実装例は以下の通りです。

```
#include <iostream>
using namespace std;

int main() {
    // 入力
    long long N;
    cin >> N;

    // 素因数分解・出力
    for (long long i = 2; i * i <= N; i++) {
        while (N % i == 0) {
            N /= i;
            cout << i << endl;
        }
    }
    if (N >= 2) cout << N << endl;
    return 0;
}
```

※ Python などのソースコードは chap3-1.md をご覧ください。