

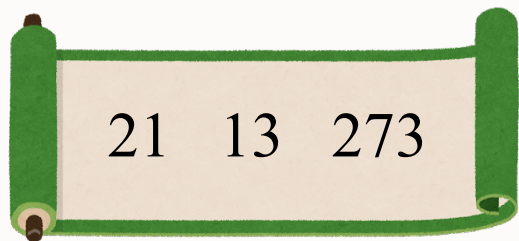
The background is a light beige color with several decorative elements: a blue circle with a dashed pattern in the top left, a pink circle in the top center, a large purple 'X' in a white circle in the top right, a wavy orange pattern in the far top right, a green and brown scroll in the bottom left, an orange plus sign in the bottom center, and a blue plus sign in the bottom right.

# JOIG 2021/2022本選 問題2 掛け算(Multiplication)

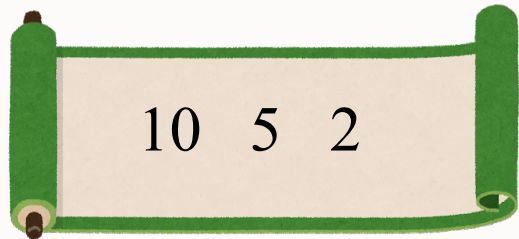
解説：@TumoiYorozu

# 問題概要

- 巻物に整数が  $N$  個書かれています。
- この中から順番を変えずに3つ選び、その整数  $x, y, z$  が  $x \times y = z$  となる選び方は何通りありますか？

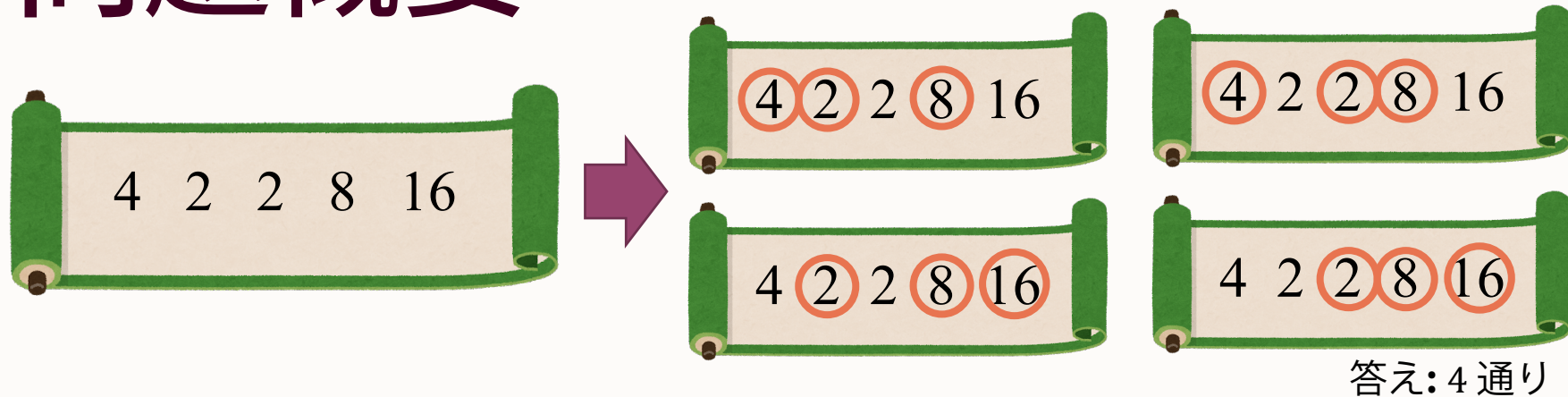


$21 \times 13 = 273$  であるので、  
条件を満たす選び方は **(21, 13, 273)** の1通りのみ



$2 \times 5 = 10$  だが並び替えてはいけないので、  
条件を満たす選び方は無い。答えは0通り

# 問題概要



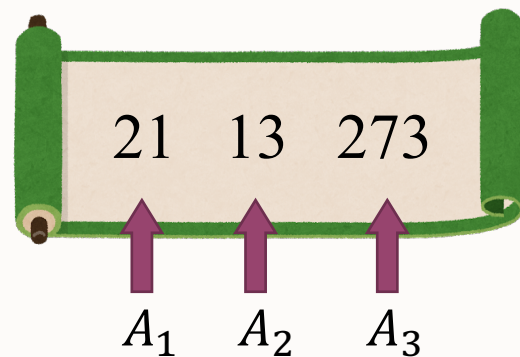
## 制約

巻物に書かれている整数の個数  $N$  が、

- $N = 3$  のとき、**20** 点
- $3 \leq N \leq 100$  のとき、更に **80** 点 (満点)

# 考察 (小課題1: $N=3$ のとき)

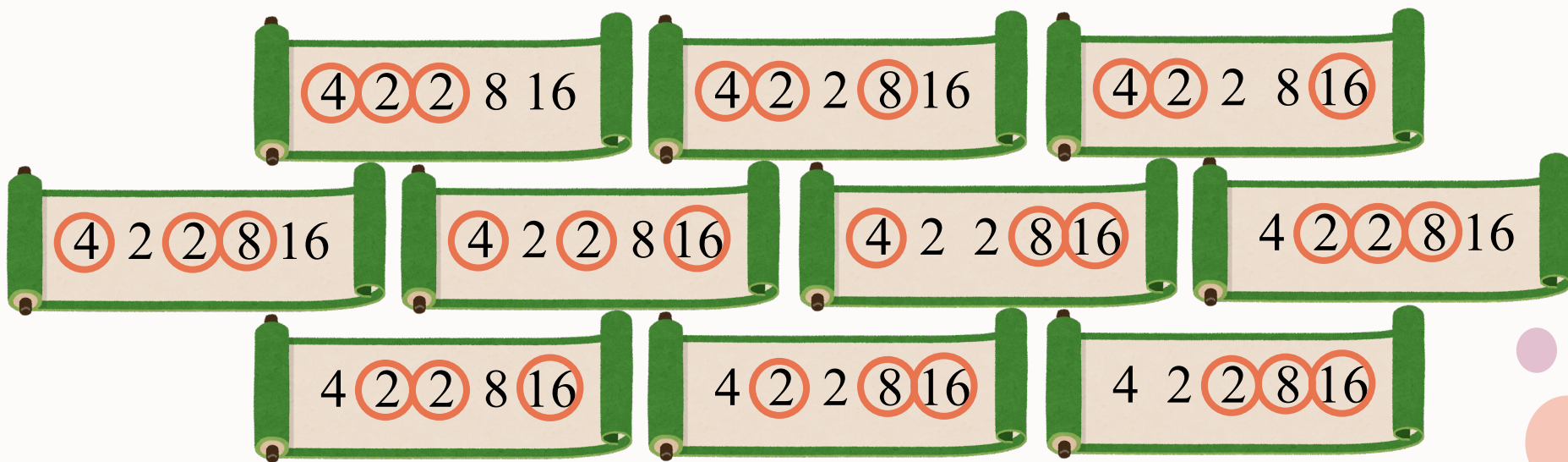
- 整数が3つしか無いので、考えるべき組み合わせは1通り
- この3つの整数  $A_1, A_2, A_3$  が、
  - $A_1 \times A_2 = A_3$  ならば、答えは1
  - $A_1 \times A_2 \neq A_3$  ならば、答えは0
- **if** などを使った条件分岐を用いることで得点することができる



# 考察 (満点: $N \leq 100$ のとき)

順番を変えずに3つ選び、その整数  $x, y, z$  が  $x \times y = z$  となる選び方は何通りありますか？

- 3つの選び方をすべて試して、条件に合うものをカウントする (全探索)
- 100個から3個を選ぶ組み合わせは**161,700**個あるが、コンピューターを使えば一瞬で調べることができる



# 全探索の仕方の考察

- 「全探索」と言っても、問題ごとにどう探索すべきかは異なる
- 手で全探索をしようとしたときを考えると、
  - ① まず1つめの数  $x$  を決めて、
  - ② 次に  $x$  より右の中から、2つめの数  $y$  を決めて、
  - ③ 次に  $y$  より右の中から、3つめの数  $z$  を決めて、
  - ④ 最後に、この選んだ3つの数  $x, y, z$  が、条件を満たすかをチェックを、順番に行う方法が考えられる



図: ①②で、 $4, 2$ の2つを決めたあと、③で3つめの数を順番に選ぶ様子

# プログラムでこれをするには？

➤ 繰り返し文を入れ子にした、多重ループで行う

## C++ での例

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

int main() {
    int N;
    cin >> N;
    int A[100];
    for (int i = 0; i < N; ++i) {
        cin >> A[i];
    }

    int answer = 0;
    for (int i = 0; i < N; ++i) {
        for (int j = i + 1; j < N; ++j) {
            for (int k = j + 1; k < N; ++k) {
                if (A[i] * A[j] == A[k]) {
                    answer += 1;
                }
            }
        }
    }

    cout << answer << endl;
}
```

## Python3での例

```
N = int(input())
A = list(map(int, input().split()))

answer = 0
for i in range(N):
    for j in range(i+1, N):
        for k in range(j+1, N):
            if A[i] * A[j] == A[k]:
                answer += 1

print(answer)
```

# 多重ループの解説

```
for (int i = 0; i < N; ++i) {  
    for (int j = i + 1; j < N; ++j) {  
        for (int k = j + 1; k < N; ++k) {  
            if (A[i] * A[j] == A[k]) {  
                answer += 1;  
            }  
        }  
    }  
}
```

C++ の例

- まず、最初の数  $x$  を選ぶ
- $i$  は何番目の数を選んだかを表す。  $x = A_i$
- $i$  を 1 番目～  $N$  番目の中で順番に試すので **for** を使って繰り返す
- プログラミングでは**0**番目から数えるので、 $i$  を**0**から始めて、  $N$  未満まで試す



# 多重ループの解説

```
for (int i = 0; i < N; ++i) {  
    for (int j = i + 1; j < N; ++j) {  
        for (int k = j + 1; k < N; ++k) {  
            if (A[i] * A[j] == A[k]) {  
                answer += 1;  
            }  
        }  
    }  
}
```

C++ での例

- 次に、2 番目の数  $y$  を選ぶ
- $j$  は何番目の数を選んだかを表す。  $y = A_j$
- $y$  は、1 番目の数  $x$  よりも後ろなので、 $j$  は  $i + 1$  から  $N$  番目の中で順番に試す

# 多重ループの解説

```
for (int i = 0; i < N; ++i) {  
    for (int j = i + 1; j < N; ++j) {  
        for (int k = j + 1; k < N; ++k) {  
            if (A[i] * A[j] == A[k]) {  
                answer += 1;  
            }  
        }  
    }  
}
```

C++ での例

- 最後に、3番目の数  $z$  を選ぶ
- $k$  は何番目の数を選んだかを表す。  $z = A_k$
- $z$  は、2番目の数  $y$  よりも後ろなので、 $k$  は  $j + 1$  から  $N$  番目の中で順番に試す

# 多重ループの解説

```
for (int i = 0; i < N; ++i) {  
    for (int j = i + 1; j < N; ++j) {  
        for (int k = j + 1; k < N; ++k) {  
            if (A[i] * A[j] == A[k]) {  
                answer += 1;  
            }  
        }  
    }  
}
```

C++ での例

- $i, j, k$  が選ばれたので、条件を満たすか **if** 文で判定し、条件に合うならば **answer** に 1 を足す

# まとめ

- コンピューターを使えば、手計算では量が多くて時間がかかる事を、高速に処理してくれる
- 複雑な処理でも、**for** や **if** などを組み合わせ、ときにはたくさん入れ子にすることで、求めることができる