

問題 3.2.1

答案如下所示。不懂的人可以回到 **3.2.2項** 確認。。

| 步驟數 | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
|------|-----|-----|-----|-----|----|----|---|
| A 的值 | 372 | 372 | 104 | 104 | 14 | 14 | 0 |
| B 的值 | 506 | 134 | 134 | 30 | 30 | 2 | 2 |

問題 3.2.2

整數 A_1, A_2, \dots, A_N 的最大公因數可以如下計算（→**3.2.5項**）

- 首先計算 A_1 和 A_2 的最大公因數。
- 接著，計算上一步的計算結果與 A_3 的最大公因數。。
- ：
- 接著，計算上一步的計算結果與 A_N 的最大公因數。

將這些步驟進行實作如下。函式 $\text{GCD}(A, B)$ 用於計算 A 和 B 的最大公因數。另外，變數 r 表示前一步的計算結果。

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

long long GCD(long long A, long long B) {
    while (A >= 1 && B >= 1) {
        if (A < B) B = B % A; // A < B 時，改寫較大的數 B
        else A = A % B; // A >= B 時，改寫較大的數 A
    }
    if (A >= 1) return A;
    return B;
}

long long N;
long long A[100009];

int main() {
```

```

// 輸入
cin >> N;
for (int i = 1; i <= N; i++) cin >> A[i];

// 求出答案
long long R = GCD(A[1], A[2]);
for (int i = 3; i <= N; i++) {
    R = GCD(R, A[i]);
}

// 輸出
cout << R << endl;
return 0;
}

```

※ Python等原始碼請參閱 chap3-2.md 。

問題 3.2.3

整數 A_1, A_2, \dots, A_N 的最小公倍數可以如下計算。

- 首先計算 A_1 和 A_2 的最小公倍數。
- 接著，計算上一步的計算結果與 A_3 的最小公倍數。。
- :
- 接著，計算上一步的計算結果與 A_N 的最小公倍數。。

此外，2 個數 A 和 B 具有以下性質（→2.5.2 項）

$$A \times B = (A \text{ 和 } B \text{ 的最大公因數}) \times (A \text{ 和 } B \text{ 的最小公倍數})$$

$$\text{即 } (A \text{ 和 } B \text{ 的最小公倍數}) = \frac{A \times B}{(A \text{ 和 } B \text{ 的最大公因數})}$$

因此，撰寫如下程式可以得出正確答案。又，函式 $\text{LCM}(A, B)$ 用於計算 A 和 B 的最小公倍數。

```

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

long long GCD(long long A, long long B) {
    while (A >= 1 && B >= 1) {
        if (A < B) B = B % A; // A < B 時，改寫較大的數 B
        else A = A % B; // A >= B 時，改寫較大的數 A
    }
}

```

```

    }
    if (A >= 1) return A;
    return B;
}

long long LCM(long long A, long long B) {
    return (A / GCD(A, B)) * B;
}

long long N;
long long A[100009];

int main() {
    // 輸入
    cin >> N;
    for (int i = 1; i <= N; i++) cin >> A[i];

    // 求出答案
    long long R = LCM(A[1], A[2]);
    for (int i = 3; i <= N; i++) {
        R = LCM(R, A[i]);
    }

    // 輸出
    cout << R << endl;
    return 0;
}

```

※ Python等原始碼請參閱 chap3-2.md。