

問題 2.5.1

この問題は、シグマ記号（→2.5.9項）の理解を問う問題です。

答えは以下の通りになります。

$$\sum_{i=1}^{100} i = (1 + 2 + 3 + \dots + 100) = 5050$$
$$\sum_{i=1}^3 \sum_{j=1}^3 ij = (1 + 2 + 3 + 2 + 4 + 6 + 3 + 6 + 9) = 36$$

なお、1 から 100 までの総和は 1.1 節にも記されていますが、和の公式（→2.5.10 項）を用いて $100 \times 101 \div 2 = 5050$ と計算することもできます。

問題 2.5.2

集合の基本（→2.5.5項）の理解を問う問題です。答えは以下のようになります。

1. $|S| = 3, |T| = 4$
2. $S \cup T = \{2, 3, 4, 7, 8, 9\}$ （一方に含まれる部分）
3. $S \cap T = \{2\}$ （両方に含まれる部分）
4. 空でない部分集合は $\{2\}, \{4\}, \{7\}, \{2, 4\}, \{2, 7\}, \{4, 7\}, \{2, 4, 7\}$ の 7 つ

分からない人は、58 ページに戻って確認しましょう。

問題 2.5.3

階乗 $N! = 1 \times 2 \times 3 \times \dots \times N$ であるため、for 文を用いて掛け算を行うプログラムを書けば良いです。なお、 $N = 20$ のとき $N! = 2.4 \times 10^{18}$ 程度となり、int 型などの 32 ビット整数ではオーバーフローを起こすことに注意してください。（次のソースコードでは、代わりに long long 型を使っています）

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

```
int main() {
    long long Answer = 1;
    for (int i = 2; i <= N; i++) {
        Answer *= i; // Answer に i を掛ける
    }
    cout << Answer << endl;
    return 0;
}
```

※ Python などのソースコードは GitHub の codes フォルダをご覧ください。

問題 2.5.4

以下のようなプログラムを書くと正解が得られます。なお、関数 `isprime(x)` は 2 以上の整数 x が素数であるかどうかを判定する関数であり、素数の場合 `true`、そうでない場合 `false` を返します。また、

- x は 2 で割り切れますか？
- x は 3 で割り切れますか？
- :
- x は $N - 1$ で割り切れますか？

といった感じで 1 つずつ調べていくことで、素数判定を行っています。

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

bool isprime(int x) {
    for (int i = 2; i <= x - 1; i++) {
        // x を i で割った余りが 0 のとき、x は i で割り切れる
        if (x % i == 0) return false;
    }
    return true;
}

int main() {
    int N, Answer = 0;
    cin >> N;
    for (int i = 2; i <= N; i++) {
        if (isprime(i) == true) cout << i << endl;
    }
    return 0;
}
```

※ Python などのソースコードは GitHub の codes フォルダをご覧ください。

問題 2.5.5

この問題の答えは **1000** です。

最も単純な方法として、 $1 \leq a \leq 4, 1 \leq b \leq 4, 1 \leq c \leq 4$ に含まれる整数の組 (a, b, c) すべてについて計算する方法がありますが、これでは面倒です。

$a = 1$ のとき
合計 100

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

$a = 2$ のとき
合計 200

2	4	6	8
4	8	12	16
6	12	18	24
8	16	24	32

$a = 3$ のとき
合計 300

3	6	9	12
6	12	18	24
9	18	27	36
12	24	36	48

$a = 4$ のとき
合計 400

4	8	12	16
8	16	24	32
12	24	36	48
16	32	48	64

そこで、以下の二重シグマの値を考えましょう。合計は **100** です。

$$\sum_{b=1}^4 \sum_{c=1}^4 bc = 100$$

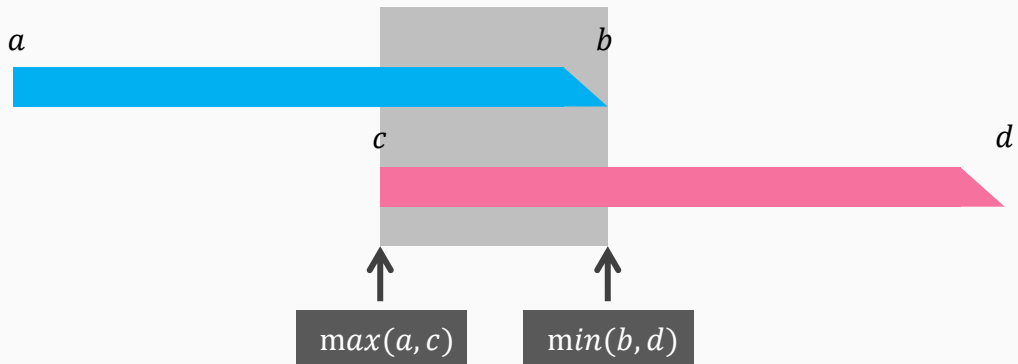
各 a における abc の合計を考えると、以下のようになります。

- $a = 1$ のときの $abc (= 1 \times bc)$ の合計: $1 \times 100 = 100$
- $a = 2$ のときの $abc (= 2 \times bc)$ の合計: $2 \times 100 = 200$
- $a = 3$ のときの $abc (= 3 \times bc)$ の合計: $3 \times 100 = 300$
- $a = 4$ のときの $abc (= 4 \times bc)$ の合計: $4 \times 100 = 400$

求める三重シグマは、上の 4 つをすべて足した値 **1000** となります。

問題 2.5.6

共通部分を持つことの必要十分条件は、 $\max(a, c) < \min(b, d)$ を満たすことです。
max 関数、min 関数が分からない人は、2.3.2 項に戻って確認しましょう。



問題 2.5.7

それぞれの i の値における「cnt が増える回数」は以下の通りです。

- $i=1$ のとき： $2 \leq j \leq N$ なので $N-1$ 回
- $i=2$ のとき： $3 \leq j \leq N$ なので $N-2$ 回
- :
- $i=N-1$ のとき：1 回
- $i=N$ のとき：0 回

よって、実行終了時の cnt の値は、和の公式（→2.5.10項）より、

$$(N-1) + (N-2) + \cdots + 1 + 0 = \frac{(N-1) \times N}{2} \left(= \frac{1}{2}N^2 - \frac{1}{2}N \right)$$

となります。cnt の値の中で最も重要な項は $(1/2) \times N^2$ であるため、このプログラムの計算量は $O(N^2)$ です。（→2.4.8項）