

1. ABC 003 A - 来月は何月？

$N = 12 \rightarrow 1, N \neq 12 \rightarrow N + 1$ と出力すればよい。

解答例

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N;
    cin >> N;
    cout << N % 12 + 1 << endl;
    return 0;
}
```

URL

https://atcoder.jp/contests/abc011/tasks/abc011_1

2. ABC 003 C - Snack

再帰関数を使ってユークリッドの互除法を行い A, B の最大公約数 G を求め、最大公約数 L は

$$AB = GL$$

の関係から求める。

解答例

```
#include <bits/stdc++.h>
using namespace std;

long long gcd(long long a, long long b) {
    if (b == 0) {
        return a;
    }
    else {
        return gcd(b, a % b);
    }
}

int main() {
    long long A, B;
    cin >> A >> B;
    cout << (A / gcd(max(A, B), min(A, B))) * B << endl;
    return 0;
}
```

URL

https://atcoder.jp/contests/abc148/tasks/abc148_c

3. ABC 055 B - Training Camp

$N!$ を 1000000007 で割った余りを求めればよい。

掛け算するたびにあまりをとる必要がある。

再帰関数を使いやすい例だがメリットはあまりない。

解答例

```
#include <bits/stdc++.h>
using namespace std;

long long f(long long n) {
    if (n == 0) {
        return 1;
    }
    else {
        return (n * f(n - 1)) % 1000000007;
    }
}

int main() {
    long long N;
    cin >> N;
    cout << f(N) << endl;
    return 0;
}
```

URL

https://atcoder.jp/contests/abc055/tasks/abc055_b

4. ABC 127 C - Prison

いもす法を使ってカードごとに開けられる扉の数を求める。

開けられる扉の数が M のものを数えればよい。

解答例

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    long long N, M, ans = 0;
    cin >> N >> M;
    vector<int> A(N + 1);
    vector<int> L(M);
    vector<int> R(M);
    for (int i = 0; i < M; i++){
        cin >> L[i] >> R[i];
        A[L[i] - 1]++;
        A[R[i]]--;
    }
    for (int i = 0; i < N; i++){
        A[i + 1] += A[i];
        if (A[i] == M){
            ans++;
        }
    }
    cout << ans << endl;
    return 0;
}
```

URL

https://atcoder.jp/contests/abc127/tasks/abc127_c

5.ABC 177 C - Sum of product of pairs

二重ループで全部の和をとると計算量 $O(N^2)$ で間に合わない。

$$\sum_{i=1}^N a_i \left\{ \sum_{j=1}^N a_j - \sum_{j=1}^i a_j \right\}$$

上記の式を用いると

計算量 $O(N)$ で解ける。

解答例

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    long long N, sum = 0, ans = 0;
    cin >> N;
    vector<long long> A(N);
    for (int i = 0; i < N; i++) {
        cin >> A[i];
        sum += A[i];
        sum %= 1000000007;
    }
    for (int i = 0; i < N; i++) {
        sum = (sum - A[i] + 1000000007) % 1000000007;
        ans += (sum * A[i]);
        ans %= 1000000007;
    }
    cout << ans << endl;
    return 0;
}
```

URL

https://atcoder.jp/contests/abc177/tasks/abc177_c

6. ABC 146 C - Buy an Integer

整数一つ一つを確かめていくと計算量 $O(N)$

買う整数が高くなるほど値段も高くなる。値段は単調増加する。

二分探索で買える、買えない、の境界を探せばよい。

解答例

```
#include <bits/stdc++.h>
using namespace std;

int main(){
    long long A, B, X, l = 0, r = 1000000001, mid;
    string s;
    cin >> A >> B >> X;
    while (l < r - 1){
        mid = (l + r) / 2;
        s = to_string(mid);
        if ((A * mid + B * s.size() <= X)){
            l = mid;
        }
        else{
            r = mid;
        }
    }
    cout << l << endl;
    return 0;
}
```

URL

https://atcoder.jp/contests/abc146/tasks/abc146_c

7. ABC 045 C - たくさんの数式

$|S| = 10$ が最大なので全ての数字の間に+を入れる入れないを考えても 512 通り。

全パターンを足し合わせる

再帰関数を使うと楽かもしれないです。

解答例

```
#include <bits/stdc++.h>
using namespace std;

long long cal(string s, long long sum, long long now, long long i) {
    now += (s[i] - '0');
    if (i == s.size() - 1) {
        return sum + now;
    }
    else {
        return cal(s, sum + now, 0, i + 1) + cal(s, sum, now * 10, i + 1);
    }
}

int main() {
    string s;
    cin >> s;
    cout << cal(s, 0, 0, 0) << endl;
    return 0;
}
```

URL

https://atcoder.jp/contests/abc045/tasks/arc061_a

8. ABC 096 D - Five, Five Everywhere

5 で割った時の余りが等しいものを 5 つ足すと 5 の倍数になる。

5 で割ると余りが 1 の素数を N 個出力すればよい。

解答例

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int N, i = 0, t = 0;
    cin >> N;
    while (t < N) {
        for (int j = 2; (j * j) <= (10 * i + 1); j++) {
            if ((10 * i + 1) % j == 0) {
                break;
            }
            if (((j + 1) * (j + 1)) > 10 * i + 1) {
                cout << 10 * i + 1 << ' ' << flush;
                t++;
            }
        }
        i++;
    }
    cout << endl;
    return 0;
}
```

URL

https://atcoder.jp/contests/abc096/tasks/abc096_d

9. 第6回 ドワゴンからの挑戦状予選 B - Fusing Slimes

右から i 番目のスライムが右から $i+j-1$ 匹目と $i+j$ 匹目のスライムの間の空間を通る確率 p_{ij} を考えると ($0 \leq j \leq N-i$)、右から i 番目のスライムと $i+j$ 匹目のスライムの間の全部のスライムが右から i 番目のスライムのより先に移動させられる確率であるから、

$$p_{ij} = \frac{1}{j+1}$$

である。これにより、移動距離の期待値は

$$\sum_{i=1}^{N-1} \left\{ (x_{i+1} - x_i) \sum_{j=0}^{i-1} \frac{1}{j+1} \right\}$$

となるので、

$$\begin{aligned} & (N-1)! \left[\sum_{i=1}^{N-1} \left\{ (x_{i+1} - x_i) \sum_{j=0}^{i-1} \frac{1}{j+1} \right\} \right] \\ &= \sum_{i=1}^{N-1} \left\{ (x_{i+1} - x_i) \sum_{j=1}^i \frac{(N-1)!}{j} \right\} \end{aligned}$$

を求めればよい。

ただし、

$$\frac{(N-1)!}{j}$$

の計算は mod 1000000007 における j の逆元を計算し $(N-1)!$ に掛ける必要がある。

解答例

```
#include <bits/stdc++.h>
using namespace std;
long long mp(long long x, long long n, long long mod) {
    long long a = 1;
    for (; n > 0; n >>= 1) {
        if (n & 1) {
            a = (a * x) % mod;
        }
        x = (x * x) % mod;
    }
    return a;
}

int main() {
    long long N, ans = 0;
    cin >> N;
    vector<long long> x(N);
    vector<long long> f(N + 1);
    vector<long long> g(N);
    f[0] = 1;
    for (int i = 0; i < N; i++) {
        cin >> x[i];
        f[i + 1] = (i + 1) * f[i];
        f[i + 1] %= 1000000007;
    }
    for (int i = 1; i < N; i++) {
        g[i] = ((f[N - 1] * mp(i, 1000000005, 1000000007)) % 1000000007);
        g[i] = (g[i] + g[i - 1]) % 1000000007;
    }
    for (int i = 0; i < N - 1; i++) {
        ans += ((x[i + 1] - x[i]) * g[i + 1]);
        ans %= 1000000007;
    }
    cout << ans << endl;
    return 0;
}
```

URL

https://atcoder.jp/contests/dwacon6th-prelims/tasks/dwacon6th_prelims_b