出前配達
Balance Board

原案・データセット：TumoiYorozu

問題文：amylase

解答：TumoiYorozu, amylase, cos, not, riantkb

解説：TumoiYorozu

問題概要

- $H \times W$のグリッドが与えられる
- 各マスに0〜9の整数値を取る重み $V_{i,j}$ がある
- 以下の式で求められる重心 (r, c) を求めよ

$$\sum_{i,j}\{V_{i,j}(i-r)\} = 0 \text{ かつ } \sum_{i,j}\{V_{i,j}(j-c)\} = 0$$

制約

- $1 \leq H, W \leq 100$
- $\sum_{i,j} V_{i,j} > 0$

解法A：数式を変形して解く

- 重心の定義式からも分かる通り、重心の水平成分と垂直成分は独立に考えられる。
- $\sum_{i,j}\{V_{i,j}(i-r)\}=0$ を満たすような r を考える
- $\sum_{i,j} i\, V_{i,j} \ -r\sum_{i,j} V_{i,j} = 0$
- $\sum_{i,j} i\, V_{i,j} \ = r\sum_{i,j} V_{i,j}$
- $\frac{\sum_{i,j} i\, V_{i,j}}{\sum_{i,j} V_{i,j}} = r$
- ここで、左辺は入力によって与えられる定数なので、
  2重ループで計算すれば r が求められる。 c も同様に求められる。

解法例A (C++)

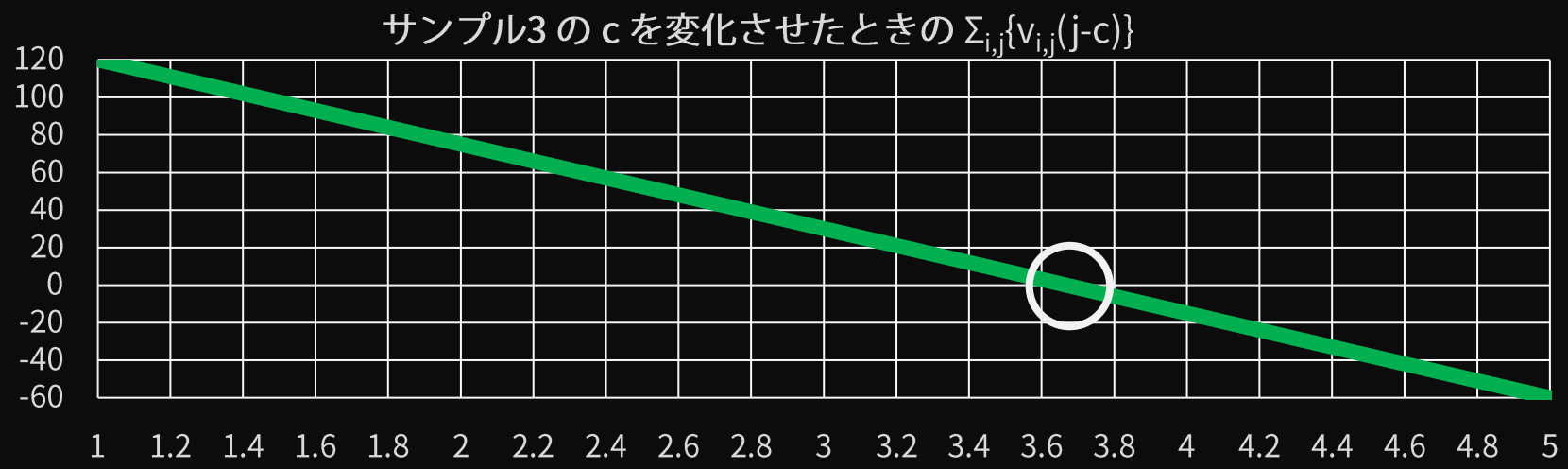
```
#include <string>
#include <iostream>
using namespace std;
int main() {
    for(;;) {
        int h, w;
        cin >> h >> w;
        if (h == 0 && w == 0) break;

        double xs = 0;
        double ys = 0;
        double a = 0;
        for (int i = 0; i < h; ++i) {
            string s;
            cin >> s;
            for (int j = 0; j < w; ++j) {
                int v = (s[j] - '0');
                xs += v * (j + 1);
                ys += v * (i + 1);
                a += v;
            }
        }
        printf("%.15f %.15f¥n", ys / a, xs / a);
    }
}
```

解法B：二分探索で解く

- r, c を変化させると、重心の式は以下のように単調減少するので、
  $[0, H]$ の範囲で r、$[0, W]$ の範囲で c の値を二分探索することにより解ける

※ちなみにこのグラフは1次関数である



解法例B (C++)

```
#include <string>
#include <iostream>
using namespace std;
int main() {
    for(;;) {
        int h, w;
        cin >> h >> w;
        if (h == 0 && w == 0) break;
        string v[100];
        for (int i = 0; i < h; ++i) {
            cin >> v[i];
        }
        double ra = 0;
        double rb = h;
        double ca = 0;
        double cb = w;
        for(int tmp = 0; tmp < 100; tmp++) {
            double r = (ra+rb)/2;
            double c = (ca+cb)/2;
            double rs = 0;
            double cs = 0;
            for (int i = 0; i < h; ++i) {
                for (int j = 0; j < w; ++j) {
                    rs += (v[i][j]-'0') * (i - r);
                    cs += (v[i][j]-'0') * (j - c);
                }
            }
            (rs > 0 ? ra : rb) = r;
            (cs > 0 ? ca : cb) = c;
        }
        printf("%.9f %.9f¥n", ra+1, cb+1);
    }
}
```

ICPC本番前のポイント

- 入力は 0~9 で表せられるが、int などの数値でなく string で受け取ると吉
- そのとき、「`-'0'`」することで、文字としての数字を整数として扱える

- 小数を出力する必要があるとき、出力の仕方も覚えておこう
  - `printf("%f", res);` や `cout << res;`
    は意図しない表示になる可能性があるのでNG
  - `printf("%.9f", res);` や `cout << fixed << setprecision(9) << res;`
    の様に、小数点以下何桁まで表示するか明示する書き方を覚えておこう

ジャッジ解

- TumoiYorozu (C++) : 35行, 757 bytes
- TumoiYorozu 2(C++) : 45行, 1003 bytes
- amylase (C++) : 37行, 763 bytes
- not (C++) : 38行, 712 bytes
- not2 (C++) : 50行, 920 bytes

統計情報

- AC / Trying Teams
  - 113/113
- First Acceptance
  - SPJ (5m05s, Guest)
  - Bu-Bu-Du-Ke (6m13s)