

電気電子情報工学実験 II (b)
実践的・競技プログラミング 練習コンテスト (Contest 1-2)
解説

廣田悠輔
y-hirota@u-fukui.ac.jp

目次

1	問題 A : 最小公倍数と最大公約数	1
2	問題 B : それにつけても金の欲しさよ	2

1 問題 A : 最小公倍数と最大公約数

解説

最初に最大公約数 B を求め、その結果を使って最小公倍数 A を求めるのが簡単である（問題名や出力の順序と逆である点に注意）。最大公約数 B の計算は Euclid の互除法により求められる。最小公倍数 A は

$$A = \frac{NM}{B}$$

で求められるので、その通りの計算を行えば良い。

解答例

```
1 #include <stdio.h>
2
3 int gcd(int n, int m) {
4     // Assume 1 <= n <= m.
5     while (m % n != 0) {
6         int p = m % n;
7         m = n;
8         n = p;
9     }
10    return n;
11 }
12
```

```

13 int main(void) {
14     int n, m;
15     int a, b; // (n, m)'s LCM, GCD.
16     scanf("%d%d", &n, &m);
17
18     b = gcd(n, m);
19     a = n / b * m;
20     printf("%d%d\n", a, b);
21
22     return 0;
23 }

```

2 問題 B : それについても金の欲しさよ

解説*1

与えられた問題文を素直に解釈すると次のように読める。

n 個の袋から m 個を選ぶ組み合わせの中で総和が最大になるものを探せ。

ところが、この解釈そのままにすべての組み合わせを調べるアルゴリズムに落とし込もうとすると、候補となる組み合わせの数が膨大になってしまう。そこで、問題文を以下のように読み替える。

n 個の袋から金額の大きい m 個を選び、その総和を求めよ。

ここまで辿り着ければ、以下の解法に自然に落とし込める。

1. 入力を読み込む。
2. 袋を金額の大きい順に並び替える (降順ソート)。
3. 金額の大きい袋から n 個について総和を計算する。

ただし、ソートについては実行時間の制限から、バブルソートなどの $O(n^2)$ のソートではなく、マージソートなどの $O(n \log n)$ のソートを使用する必要がある。

解答例

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int descending_int(const void *a, const void *b) {
5     return *((int *)b) - *((int *)a);
6 }

```

*1 ここでは最小限の説明のみを行う。詳細については次回に「競技プログラミングの問題を解くための考え方」で述べる。

```
7
8 int main(void) {
9     int n, m;
10    int *a;
11    int i;
12    int sum;
13
14    scanf("%d", &n);
15    scanf("%d", &m);
16    a = malloc(sizeof(int) * n);
17    for (i = 0; i < n; ++i) {
18        scanf("%d", &a[i]);
19    }
20
21    qsort(a, n, sizeof(int), descending_int);
22    sum = 0;
23    for (i = 0; i < m; ++i) {
24        sum += a[i];
25    }
26    printf("%d\n", sum);
27
28    free(a);
29    return 0;
30 }
```