

《数据结构与算法》第一周作业反馈

助教-陈孙一硕

1. 交换可以直接用 C++ 内置的 swap 函数来实现；
2. 求 m, n 的最小公倍数的时候，最佳做法应当是 $m/gcd(m, n) * n$ 而不是 $m * n/gcd(m, n)$ ，避免乘积超出范围限制；
3. 有同学直接用系统内置的函数 $gcd(a, b)$ 来求最大公约数，但还是希望大家能够掌握欧几里得算法的原理；
4. 部分同学还不会使用多行输入的语法：

```
while (cin>>x>>y)
```

5. 第一题取最大值的时候，很多同学会把答案的初值设置为 0，这会导致最后一个测试点无法通过。这里有一个误区，就是很容易错认为答案一定是一个非负数。这里为大家提供一组反例：-99 100。按照上面这种错误的思路，就会输出 0。然而事实上，答案应该是-1。

下面附上我写的这几道题的代码：

1. 练习题

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

signed main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int x, y;
    while (cin >> x >> y) {
        int ans = -1e9;
        if (x + y <= 100) {
            ans = max(ans, x + y);
        }
        if (x - y <= 100) {
```

```

        ans = max(ans, x - y);
    }
    if (y - x <= 100) {
        ans = max(ans, y - x);
    }
    if (x * y <= 100) {
        ans = max(ans, x * y);
    }
    cout << ans << endl;
}
return 0;
}

```

2. 最大公约数和最小公倍数

```

#include <bits/stdc++.h>
using namespace std;
#define int long long

int gcd(int x, int y) {
    if (x < y) {
        swap(x, y);
    }
    if (x % y == 0) {
        return y;
    }
    return gcd(y, x % y);
}

signed main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int m, n;
    cin >> m >> n;
    cout << gcd(m, n) << " " << m / gcd(m, n)*n << endl;
    return 0;
}

```

3. 最大公约数（限时）

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

int gcd(int x, int y) {
    if (x < y) {
        swap(x, y);
    }
    if (x % y == 0) {
        return y;
    }
    return gcd(y, x % y);
}

signed main() {
    ios::sync_with_stdio(false);
    cin.tie(nullptr);
    int m, n;
    cin >> m >> n;
    cout << gcd(m, n) << endl;
    return 0;
}
```