

• A

- 题意：给一个数 n 进行 k 次运算 运算规则，得到 n 除了 1 之外能够整除 n 的最小数字，然后将结果 $f(n)$ 加到 n 上，在经过 k 次运算，得到最终结果
- 思路：将 n 分为偶数和奇数两部分，如果 n 为偶数， $f(n) = 2 \mid n + f(n)$ 也是偶数，之后的迭代就直接加 2 就行了，如果 n 为奇数，有两种情况，第一，他为质数，第二，他的 $f(n)$ 为质数，但是无论如何，他在一次迭代后得到的数都是偶数，然后再用一的情况就行了。
- 代码：

```

#include <bits/stdc++.h>
using namespace std;
int sdiv(int n){
    for(int i = 2; i <= sqrt(n); i++){
        if(n % i == 0) return i;
    }
    return n;
}
int main(){
    int n;
    scanf("%d",&n);
    while(n--){
        int k, t;
        scanf("%d%d",&k,&t);
        if(sdiv(k) % 2 == 1){
            k += sdiv(k);
            k += (t-1)*2;
        }
        else
        {
            k += t * 2;
        }
        printf("%d\n",k);
    }
    return 0;
}

```

• B

- 题意：给你一串数 $s_1, s_2, s_3, \dots, s_n$ 。如果满足下标 $i, i+1$ 满足 $s_i < s_{i+1}$ 并且有 $s_{i+1} < s_{i+2}$ 。则称这样的安排是美的，题目要你找出一串序列中最长的，具有美感的数列长度。

◦ 题解：不多bb

◦ 代码：

```
# include <bits/stdc++.h>

using namespace std;

void solve() {
    int n;
    cin >> n;
    vector <int> a(n + 1), dp(n + 1);
    for (int i = 1; i <= n; i++) {
        cin >> a[i];
    }
    int answer = 0;
    for (int i = 1; i <= n; i++) {
        int mx = 0;
        for (int j = 1; j * j <= i; j++) {
            if (i % j == 0) {
                if (a[j] < a[i]) mx = max(mx, dp[j]);
                if (a[i / j] < a[i]) mx = max(mx, dp[i / j]);
            }
        }
        dp[i] = mx + 1;
        answer = max(answer, dp[i]);
    }
    cout << answer << endl;
}

int main() {
    int tt = 1;
    cin >> tt;
    while (tt--)
        solve();
}
```

• C

◦ 题意：给你一串数 $s_1, s_2, s_3, \dots, s_n$ 。先算这串数的lcm再算这串数的gcd

◦ 题解：开始没仔细思考就直接暴力，直接按题意来算，提交后就后悔了，暴力肯定会超时的啊，后来想到了STL大法，然后就过了。

- 代码：

```
#include <bits/stdc++.h>

using namespace std;

int main() {
    int n;
    cin >> n;
    vector<int> a(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i];
    }
    vector<int> pref(n + 1);
    for (int i = 0; i < n; i++) {
        pref[i + 1] = __gcd(pref[i], a[i]);
    }
    vector<int> suf(n + 1);
    for (int i = n - 1; i >= 0; i--) {
        suf[i] = __gcd(suf[i + 1], a[i]);
    }
    long long ans = 1;
    for (int i = 0; i < n; i++) {
        int x = __gcd(pref[i], suf[i + 1]);
        ans = ans / __gcd(ans, (long long) x) * x;
    }
    cout << ans << '\n';
    return 0;
}
```

- D

- 题目 <https://codeforces.com/contest/1350/problem/D>
- 题解：哦时间不够了，没做完，提交了也没过。
- 咕咕中，会补的，别催了。

- E

- 题目 <https://codeforces.com/contest/1350/problem/E>


- 咕咕中，会补的，别催了。

- F

- 题目 <https://codeforces.com/contest/1350/problem/F>

- 咕咕中，会补的，别催了。

From:
<https://wiki.cvbbacm.com/> - CVBB ACM Team

Permanent link:
https://wiki.cvbbacm.com/doku.php?id=2020-2021:teams:manespace:codeforces_642_div2&rev=1589534627 

Last update: **2020/05/15 17:23**