

A 绘制日历

这个题目需要注意格式，题目已经说明得比较详细了，编码时注意即可

code:

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  const int maxn = 1e5 + 7;
6  int month_day[13] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
7  int days[13] = {};
8  void print_calendar(int w, int m)
9  {
10     //先计算这个月一号是周几，之后输出
11
12     //计算这个月1号是周几
13     int now_day_in_week = (days[m - 1] + w - 1) % 7 + 1;
14     for (int i = 0; i < now_day_in_week - 1; i++)
15     {
16         printf(" ");
17     }
18     //这一天是周几
19     int day_in_week = now_day_in_week;
20     for (int i = 1; i <= month_day[m]; i++)
21     {
22         printf("%2d", i);
23         if (day_in_week == 7 && i != month_day[m])
24             puts("");
25         else if (i != month_day[m])
26             printf(" ");
27         day_in_week += 1;
28         if (day_in_week == 8)
29             day_in_week = 1;
30     }
31     puts("");
32 }
33 int main()
34 {
35     for (int i = 1; i < 13; i++)
36     {
37         days[i] = month_day[i] + days[i - 1];
38     }
39     int N, w, m;
40     cin >> N;
41
42     for (int i = 0; i < N; i++)
43     {
44         cin >> w >> m;
45         print_calendar(w, m);
46     }
47
48     return 0;
49 }
50
```

B 高难度运算

这一题涉及到位运算，有个简单的知识点就是 $(a \& b) + (a | b) = a + b$

知道这一点后，这题就没什么难度了。当然，也可以找一下规律，不难发现。

C 你画我猜

简单的签到题，注意格式就行。

D 双十一满减

简单的签到题，考察一下浮点数

E 只办三件事

考察一下大家会不会读入整行

F 最大的字符串长度乘积

枚举一下字符串就行了，注意顺序

G offer收割机

用一下快速排序就能解决

H 出题人的密码困境

主要是一点：如何判断重复的PIN，方法有很多，下面代码给出一种。

```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  const int maxn = 1e5 + 7;
6  int p[maxn];
7  vector<int> ans;
8  int my_map[10000] = {};
9  int main()
10 {
11     int n;
12     int T;
13     cin >> T;
14     while (T--)
15     {
16         cin >> n;
17         ans.clear();
18         for (int i = 0; i < 10000; i++)
19             my_map[i] = 0;
20         for (int i = 0; i < n; i++)
21         {
22             cin >> p[i];
23             my_map[p[i]]++;
24         }
25
26         for (int i = 0; i < n; i++)
27         {
28             if (my_map[p[i]] == 1)
29                 ans.push_back(p[i]);
30         }
31         if (ans.size())
32         {
33             int sz = ans.size();
34             printf("%d\n%04d", sz, ans[0]);
35             for (int i = 1; i < sz; i++)
36             {
```

```

37         printf(" %04d", ans[i]);
38     }
39 }
40 else
41 {
42     printf("oh my pins");
43 }
44 puts("");
45 }
46 return 0;
47 }

```

l ksk

究极签到题

J 素数问题

稍微有点意思的题目，一开始数据出了点问题，很抱歉=，=

分析：

这道题，首先观察到样例数量比较多，那么就应该考虑每个样例不能花费太多时间计算。另外，根据题目要求，可以提前计算出来首尾一致的素数，之后每个样例进行查询即可。这种方法也就是“打表”，是一种比较常用的技巧。当然，每个样例的查询也要尽量快，可以用二分查找加快速度。

代码：

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  typedef long long ll;
5  const int maxn = 1e5 + 7;
6  bool pd(int x)
7  {
8      int sqx = sqrt(x);
9      if (x < 2)
10         return false;
11
12     for (int i = 2; i <= sqx; i++)
13     {
14         if (x % i == 0)
15             return false;
16     }
17     return true;
18 }
19 bool pd_same(int x)
20 {
21     int last_num = x % 10;
22     while (x >= 10)
23     {
24         x /= 10;
25     }
26     int first_num = x;
27     return first_num == last_num;
28 }
29 int biao[2000];
30 int main()
31 {
32     int biao_sz = 0;
33     // 打表记录满足要求的素数

```

```
34     for (int i = 0; i <= 100000; i++)
35     {
36         if (pd(i) && pd_same(i))
37         {
38             biao[biao_sz++] = i;
39         }
40     }
41     int T, a, b;
42     cin >> T;
43     while (T--)
44     {
45         scanf("%d%d", &a, &b);
46         int ans = upper_bound(biao, biao + biao_sz, b) - lower_bound(biao,
47 biao + biao_sz, a);
47         printf("%d\n",ans);
48     }
49     return 0;
50 }
```