

第7次随机赛(Div4)

一：找单词游戏

分析：

读入一个字符串，然后遇到逗号，句号，空格的时候跳过，并且记录一下上一个单词是谁，以及长度，找到最长的就好。在我下面的写法中特殊判断了 $i \neq 0$ 是因为如果遇到多个空格的时候，长度会被清空成 0

AC Code

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

using namespace std;

int main(){
    string s;
    getline(cin, s);
    int n = s.size();
    s.insert(0, " ");
    s = s + " ";
    int len = 0;
    string a;
    int maxn = -1;
    int minn = 110;
    string ans_max, ans_min;
    for(int i = 1; i <= n + 1; i++){
        if(s[i] == ' ' || s[i] == ',' || s[i] == '.'){
            if(len > maxn && len != 0){
                maxn = len;
                ans_max = a;
            }
            if(len < minn && len != 0){
                minn = len;
                ans_min = a;
            }
            len = 0;
            a.clear();
            continue;
        }
        a += s[i];
        len++;
    }
    cout << ans_max << endl << ans_min;
}
```

二：工作日

分析:

题目数据是 10^{18} 首先应该开 `long long` , 其次, 我们知道一周有5个工作日, 所以答案就是 $ans = \lfloor \frac{n}{7} \rfloor$, 其次我们可能还会剩下 `n mod 7` 天, 暴力循环确定这 7 天即可。注意如果是采取取余的方式判断, 注意要考虑 `7 % 7` 的情况。

AC Coder

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

using namespace std;

int main(){
    int x;
    long long n;
    cin >> x >> n;
    long long ans = 0;
    ans += n / 7 * 5;
    n %= 7;
    for(int i = 1; i <= n; i ++ ){
        if(((i + x - 1) % 7 == 0 ? 7 : (i + x - 1)) != 7 && ((i + x - 1) % 7) !=
6){
            //cout << i << " " << x << endl;
            ans ++;
        }
    }
    cout << ans;
}
```

三: 到底去不去重!

分析:

题目只是说了去重, 没有要求按照顺序输出, 即不可以使用 `set` 函数。考虑使用哈希表, 数组模拟即可。 `a[i] = 0` 表示 `i` 这个数字没有出现, `a[i] = 1` 表示 `i` 这个数字出现过, 所以如果没有出现过我们就输出这个数字, 并且把它改为 1 即可。

AC Code

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

using namespace std;

int b[10010];

int main(){
    int n;
    cin >> n;
    for(int i = 1; i <= n; i ++ ){
        int x;
        cin >> x;
        if(b[x] == 0){
            cout << x << " ";
            b[x] = 1;
        }
    }
}
```

```
}
```

四： 动态规划？最长上升子序列？

分析：

题目求的是最长连续的序列。

并且需要满足

$$a_i < a_{i+1} < a_{i+2} < \dots < a_{j-1} < a_j$$

所以我们考虑使用

- pre_i 表示 $1 \sim i$ 中小于 i 的连续子序列长度
- nxt_i 表示 $i \sim n$ 中小于 i 的连续子序列长度

对于 pre 可以正着枚举

对于 nxt 可以反着枚举

最后答案是 $pre_i + nxt_i - 1$ 最后的 -1 是因为我们的表示中包括了 i 自己，所以算的时候会重复计算，如果你的状态是 $1 \sim n - 1$ 以及 $i + 1 \sim n$ 的话 则不需要，但是需要 $+1$ 即可。

AC Coder

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

using namespace std;

int a[1000010];
int pre[1000010];
int nxt[1000010];

int main(){
    int n;
    scanf("%d", &n);
    for(int i = 1; i <= n; i ++ ){
        scanf("%d", a + i);
    }
    pre[1] = 1;
    for(int i = 2; i <= n; i ++ ){
        if(a[i] > a[i - 1]){
            pre[i] = pre[i - 1] + 1;
        } else {
            pre[i] = 1;
        }
    }
    nxt[n] = 1;
    for(int i = n - 1; i >= 1; i -- ){
        if(a[i] > a[i + 1]){
            nxt[i] = nxt[i + 1] + 1;
        } else {
            nxt[i] = 1;
        }
    }
}
```

```
int ans = -1;
for(int i = 1; i <= n; i ++ ){
    ans = max(ans, pre[i] + nxt[i] - 1);
}
cout << ans;
}
```

五: 小学生! 检查作业!

分析:

做法很多, 字符串分割即可。但是没必要, 我们把 + 和 = 表示成字符, 输入的时候用 char 类型即可。

这样就相当于判断 $a + b = c$ 的问题了。就简单了, 如果 $a + b \neq c$ 答案 + 1

AC Coder

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

using namespace std;

int main(){
    int n;
    cin >> n;
    int ans = 0;
    for(int i = 1; i <= 3 * n; i ++ ){
        int a, b, c;
        char d, e;
        cin >> a >> d >> b >> e >> c;
        if(a + b != c){
            ans ++;
        }
    }
    cout << ans;
}
```