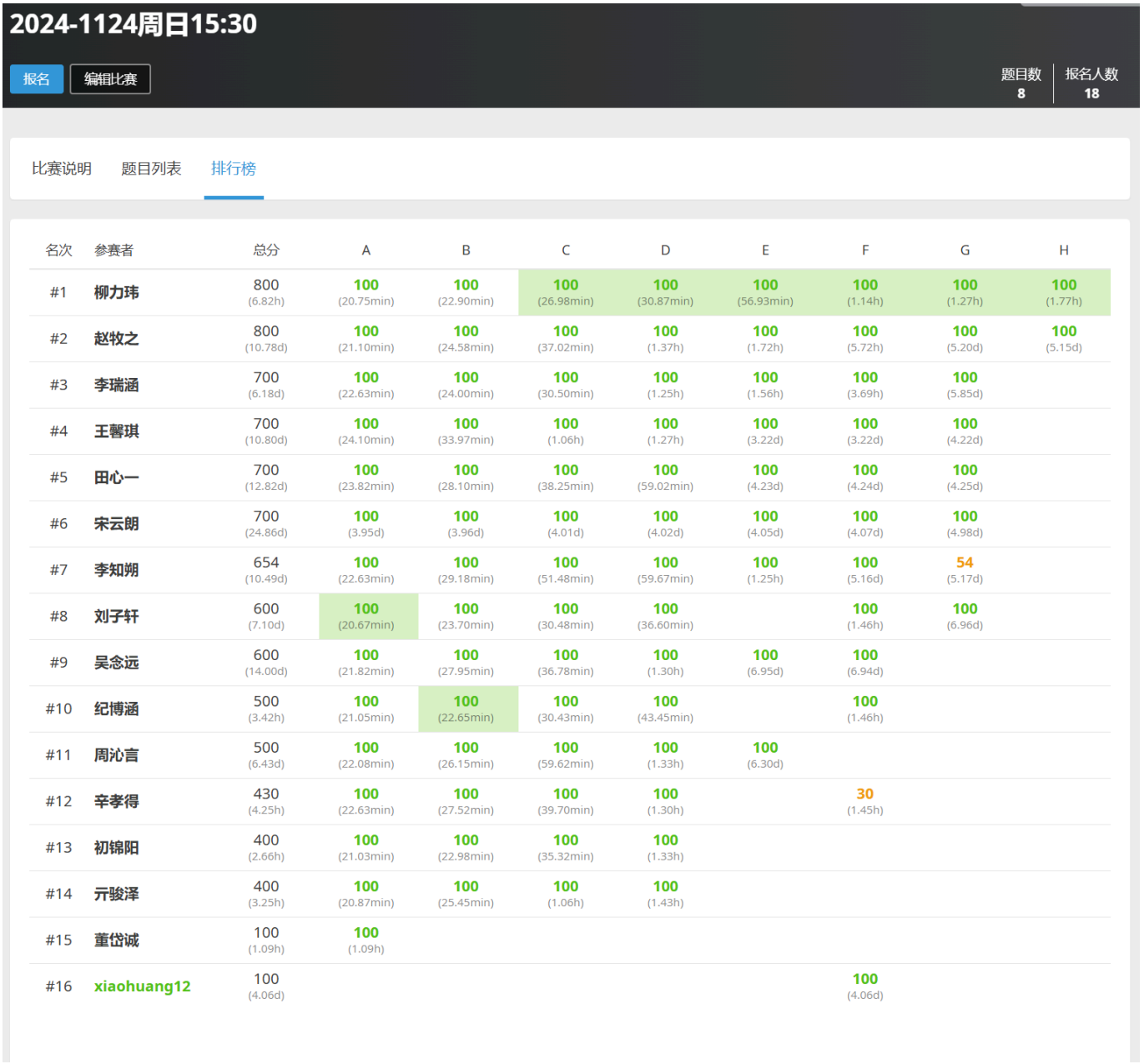
string常用方法

人员

初锦阳、田心一、吴念远、纪博涵、董岱诚、赵牧之、周沁言、亓骏泽、李知朔、李瑞涵、柳力玮、刘子轩、王馨琪 到课

上周作业检查

上周作业链接: https://www.luogu.com.cn/contest/215886



2024-1124周日15:30

报名 编辑比赛

题目数 8 | 报名人数 18

比赛说明 题目列表 排行榜

名次	参赛者	总分	A	B	C	D	E	F	G	H
#1	柳力玮	800 (6.82h)	100 (20.75min)	100 (22.90min)	100 (26.98min)	100 (30.87min)	100 (56.93min)	100 (1.14h)	100 (1.27h)	100 (1.77h)
#2	赵牧之	800 (10.78d)	100 (21.10min)	100 (24.58min)	100 (37.02min)	100 (1.37h)	100 (1.72h)	100 (5.72h)	100 (5.20d)	100 (5.15d)
#3	李瑞涵	700 (6.18d)	100 (22.63min)	100 (24.00min)	100 (30.50min)	100 (1.25h)	100 (1.56h)	100 (3.69h)	100 (5.85d)	
#4	王馨琪	700 (10.80d)	100 (24.10min)	100 (33.97min)	100 (1.06h)	100 (1.27h)	100 (3.22d)	100 (3.22d)	100 (4.22d)	
#5	田心一	700 (12.82d)	100 (23.82min)	100 (28.10min)	100 (38.25min)	100 (59.02min)	100 (4.23d)	100 (4.24d)	100 (4.25d)	
#6	宋云朗	700 (24.86d)	100 (3.95d)	100 (3.96d)	100 (4.01d)	100 (4.02d)	100 (4.05d)	100 (4.07d)	100 (4.98d)	
#7	李知朔	654 (10.49d)	100 (22.63min)	100 (29.18min)	100 (51.48min)	100 (59.67min)	100 (1.25h)	100 (5.16d)	54 (5.17d)	
#8	刘子轩	600 (7.10d)	100 (20.67min)	100 (23.70min)	100 (30.48min)	100 (36.60min)		100 (1.46h)	100 (6.96d)	
#9	吴念远	600 (14.00d)	100 (21.82min)	100 (27.95min)	100 (36.78min)	100 (1.30h)	100 (6.95d)	100 (6.94d)		
#10	纪博涵	500 (3.42h)	100 (21.05min)	100 (22.65min)	100 (30.43min)	100 (43.45min)		100 (1.46h)		
#11	周沁言	500 (6.43d)	100 (22.08min)	100 (26.15min)	100 (59.62min)	100 (1.33h)	100 (6.30d)			
#12	辛孝得	430 (4.25h)	100 (22.63min)	100 (27.52min)	100 (39.70min)	100 (1.30h)		30 (1.45h)		
#13	初锦阳	400 (2.66h)	100 (21.03min)	100 (22.98min)	100 (35.32min)	100 (1.33h)				
#14	亓骏泽	400 (3.25h)	100 (20.87min)	100 (25.45min)	100 (1.06h)	100 (1.43h)				
#15	董岱诚	100 (1.09h)	100 (1.09h)							
#16	xiaohuang12	100 (4.06d)						100 (4.06d)		

作业

https://www.luogu.com.cn/contest/217655

课堂表现

同学们上课听讲做题整体都很认真，今天题目中的 A 题大部分同学做的都比较吃力一些，课下要再花时间做一下这个题。

课堂内容

U506935 保留整数

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

using namespace std;

const int maxn = 80 + 5;
char s[maxn];

int main()
{
    cin >> (s+1);
    int n = strlen(s+1);
    for (int i = 1; i <= n; i++) {
        if (s[i] 这一位是 0~9) {
            cout << s[i];
        } else {
            这个字符是一段非 0~9 的第一个
            1. 他前面是一个 0~9   s[i-1]>='0'&&s[i-1]<='9'
            2. 这是第一个字符     i==1
            两个条件通过 或者 连接
        }
    }
    return 0;
}
```

[ABC220C] Long Sequence

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

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
int w[maxn];

int main()
{
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) cin >> w[i];
  LL x; cin >> x;

  LL sum = 0;
  for (int i = 1; i <= n; ++i) sum += w[i];
```

```
  LL k = x / sum;
  LL t = k * sum;
  for (int i = 0; i <= n; ++i) {
    t += w[i];
    if (t > x) {
      cout << k*n+i << endl;
      break;
    }
  }
  return 0;
}
```

abs, swap, min, max 等方法

```
1. abs: 求绝对值

2. swap: 交换  ->  swap(a, b)

3. min: 取最小值  ->  min(a, b)

4. max: 取最大值  ->  max(a, b)

int maxx = -100000000;
for (int i = 1; i <= n; i++) {
    maxx = max(maxx, a[i]);
}

int minn = 100000000;
for (int i = 1; i <= n; i++) {
    minn = min(minn, a[i]);
}
```

U495815 1234567890

```
#include <iostream>

using namespace std;

int main()
{
    int n;
    cin >> n;
    for (int i = 1; i <= n; i++) {
        string s;
        cin >> s;  // s[0], s[1], s[2], s[3]
        int a = s[0]-'0', b = s[1]-'0', c = s[2]-'0', d = s[3]-'0';
        if (a == 0) {
```

```
            a = 10;
        }
        if (b == 0) {
            b = 10;
        }
        if (c == 0) {
            c = 10;
        }
        if (d == 0) {
            d = 10;
        }

        // 1 -> a, a -> b, b -> c, c -> d, +4
        int sum = abs(a-1) + abs(b-a) + abs(c-b) + abs(d-c) + 4;
        cout << sum << endl;
    }
    return 0;
}
```

string 的新方法

1. reverse
2. find
3. rfind
4. substr

```
string s;
reverse(s.begin(), s.end());  // 翻转 s
s.find("abc");  // 在 s 中从左到右寻找 "abc" 字符串，如果找到，返回首字母下标；如果没找到，返回 string::npos，即 int 下的 -1
s.rfind("abc");  // 在 s 中从右到左寻找 "abc" 字符串
string t = s.substr(7,3);  // 从 s 的位置 7 开始，截取长度为 3 的字符串
string t = s.substr(7);  // 从 s 的位置 7 开始，截取剩余的全部字符串部分

// 判断字符串 a 里面是否包含字符串 b
if (a.find(b) != string::npos)
if ((int)a.find(b) != -1)
```

U498471 验证子串

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

using namespace std;

int main()
{
```

```
    string s, t;
    cin >> s >> t;
    if (s.find(t) != string::npos) cout << t << " is substring of " << s << endl;
    else if (t.find(s) != string::npos) cout << s << " is substring of " << t <<
endl;
    else cout << "No substring" << endl;
    return 0;
}
```