

第十三届趣味C预赛 Day2 Solution by MSC@ZJU

基本信息

- 预计难度: 7-1 < 7-2 < 7-3 < 7-4 < 7-5
- 实际通过人数: 7-1 > 7-3 > 7-4 > 7-5 > 7-2
- 提交&通过情况
 -

标号	标题	分数	通过数	提交数	通过率
7-1	决赛日期	100	220	369	0.60
7-2	图灵测试	100	34	1264	0.03
7-3	约翰的翳病	100	107	454	0.24
7-4	玩游戏	100	44	772	0.06
7-5	婷婷的桃子	100	40	394	0.10

图灵测试

题目大意

给出一段对话判断Stranger是不是人类

解题思路

思考人类本质。每次读入两行字符串去掉名称后比较是否完全相同——是否构成复读机（似乎大家对于复读的标准定义不太一样，为出题人考虑不周抱歉）出题人没有预料到字符串处理给大家造成这么大的困扰，又因为PTA实时看分的机制不能设置很多测试点，很抱歉影响了参赛体验。

标程

```
1 #include <stdio.h>
2 #include <string.h>
3 char s1[1005], s2[1005], tmp[1005];
4 int main(){
5     int cas, n, i; scanf("%d", &cas);
6     while(cas--){
7         scanf("%d", &n);
8         int flag = 1;
9         for(i = 1; i <= n / 2; ++i){
10             scanf("%s", tmp); gets(s1);
11             scanf("%s", tmp); gets(s2);
```

```

12         if(strcmp(s1, s2) != 0) flag = 0;
13     }
14     if(flag) puts("Human-being");
15     else puts("Robot");
16 }
17 return 0;
18 }

```

并提供另一种读入方式的思路...

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define FORP(i,a,b) for(i=(a);i<=(b);i++)
5  #define maxn 100005
6  /*=====split line=====*/
7
8  char s[3][maxn];
9  int cnt[3];
10 int main(){
11     int cas; scanf("%d",&cas);
12     while (cas--){
13         int n,i,j; scanf("%d\n",&n);
14         int ff = 1;
15         FORP(j, 1, n / 2){
16             cnt[1] = 0, cnt[2] = 0;
17             FORP(i, 1, 2) {
18                 while ((s[i][++cnt[i]] = getchar()) != '\n'); cnt[i]--;
19             }
20             if (cnt[1] - 1 != cnt[2]) ff = 0;
21             FORP(i, 10, cnt[1]) if (s[1][i] != s[2][i - 1]) ff = 0;
22         }
23         if (ff) puts("Human-being");
24         else puts("Robot");
25     }
26 }

```

约翰的眼翳

题目大意

对于读入的 n 组单词，判断是否满足不存在相邻的不同字母这一条件。

解题思路

按照题意模拟即可。

数据点0为样例；

数据点1为存在单个字母的情况；

数据点2为所有可能情况。

标程

```
1  #include<stdio.h>
2  #include<string.h>
3  int check(char *str1,char* str2){
4      if (strlen(str1) != strlen(str2)) return 0;
5      if (strlen(str1) == 1) return 1;
6      for(int i = 1; str1[i] != 0; i++){
7          if (str1[i] != str2[i] &&
8              str1[i-1] != str2[i-1])
9              return 0;
10     }
11     return 1;
12 }
13
14 int n;
15 char str1[10000],str2[10000];
16 int sum;
17 int main(){
18     scanf("%d", &n);
19     for(int i = 0; i < n; i++){
20         scanf("%s", str1);
21         scanf("%s", str2);
22         if (check(str1, str2)) sum++;
23     }
24     printf("%d\n", sum);
25 }
```

玩游戏

题目大意

n 个连在一起的房子，走进某个房子可能会加血掉血或获得一枚金币，求在血量始终大于0地走过一段房子能够获得的最大金币数量

解题思路

首先对于部分分，没有T和只有T都是可以直接输答案的 $n \leq 1000$ 的情况 可以验证每一段能够获得的金币数进行比较。设置 $n \leq 10^5$ 和 $n \leq 10^6$ 是为了防止同学用 $O(n^2)$ 带一些奇怪的优化卡过去，并鼓励大家思考优美的做法...

这道题应该有很多做法，但是其实这个问题最简单的做法是从头到尾按照题意模拟，一旦血量下降为0时在这个点上重置血量和当前所获得的金币数，走到尾即可。

为什么这么做是对的呢？我们假设最优方案的起点不是上述路径中的某一个起点，那么它必然被包含在上述做法求出的某一条路径中。由于游戏要求血量大于1时才能继续进行，所以原路径肯定不会比当前路径劣，因此上述做法是正确的

标程

```
1  #include <stdio.h>
2  #include <stdlib.h>
```

```

3  int max(int a, int b){return a > b ? a : b;}
4  int main(){
5      int n; scanf("%d\n",&n);
6      int i, ans = 0;
7      int hp = 1, sum = 0;
8      for (i = 1; i <= n; i++){
9          char c = getchar();
10         switch(c){
11             case 'A': hp++; break;
12             case 'T': sum++; break;
13             case 'S': hp--; break;
14         }
15         if (hp == 0){
16             ans = max(ans, sum); hp = 1, sum = 0;
17         }
18     }
19     ans = max(ans, sum);
20     printf("%d\n", ans);
21 }

```

婷婷的桃子

题目大意

给定n个正整数，求去掉一个数后，剩下数的gcd的最大值。

解题思路

40分做法：枚举去掉哪一堆桃子，每次对剩下的桃子用辗转相除法计算最大公因数，答案取Max。100分做法：考虑线性做法。我们知道计算gcd的顺序对最终的结果是没有任何影响的，因此采用前缀/后缀思想。left[i]表示前i个数的最大公因数，right[i]表示第i个数到第n个数的最大公因数。所以只需要一次遍历，枚举去掉哪一堆桃子，通过gcd(left[i-1],right[i+1]) 便可以快速计算出剩下桃子的最大公因数。

标程

```

1  #include<stdio.h>
2  #define mid ((l+r)>>1)
3  #define ll long long
4  #define maxn 100010
5  int a[maxn],left[maxn],right[maxn];
6  int gcd(int a,int b){
7      if(b==0)return a;
8      return gcd(b,a%b);
9  }
10 int max(int a,int b){
11     if(a>b)return a;else return b;
12 }
13 int main(){
14     int n,cas;
15     scanf("%d",&cas);
16     while(cas--){

```

```
17     scanf("%d",&n);
18     for(int i=1;i<=n;i++){
19         scanf("%d",&a[i]);
20     }
21     left[1]=a[1];
22     right[n]=a[n];
23     for(int i=2;i<=n;i++)left[i]=gcd(left[i-1],a[i]);
24     for(int i=n-1;i>=1;i--)right[i]=gcd(right[i+1],a[i]);
25     int ans=max(right[2],left[n-1]);
26     for(int i=2;i<n;i++){ans=max(ans,gcd(left[i-1],right[i+1]));}
27     printf("%d\n",ans);
28 }
29 }
```

再次感谢大家的参与!