

2019年厦门大学铃盛杯C语言积分赛第二周周赛题解

A题：小龙与问号

出题人：XYC

难度评价： **Very Easy**

解法：

判断一个数能否被3整除，只要判断这个数各个位上的数字之和能否被3整除即可。

std:

```
#include <stdio.h>
#include <ctype.h>

int main() {
    char ch;
    int sum = 0;
    while ((ch = getchar()) != EOF) {
        if (isdigit(ch)) {
            sum += (ch - '0');
        }
    }
    if (sum % 3 == 0) printf("Yes");
    else printf("No");
    return 0;
}
```

B. 搞项目

出题人:LRL

难度评价： **Easy**

解法：

我们观察到，只有出现NW、NE、SW、SE才会出现多种解读，如NW可以解读为N、W和NW。那么根据乘法原理，每出现上述四个任意一个都会使得解读数*2。

注意对691787取模。取模最好乘一次就取一次模数，否则很可能会溢出。

注意到某位同学超时了12发，还把自己昵称都改了，但是你和AC只差一个strlen的距离。（请自行打开热身赛C题题解，复习strlen的使用提示）

std:

```
#include <stdio.h>

int main()
{
    int t, mod = 691787;
    char ch, las;
    scanf("%d ", &t);
    while(t--)
    {
        int sum = 1;
        ch = getchar();
        while(ch != '\n')
        {
            las = ch;
            ch = getchar();
            if((las == 'S' || las == 'N') && (ch == 'E' || ch == 'W'))
                sum = sum * 2 % mod;
        }
        printf("%d\n", sum);
    }

    return 0;
}
```

C: Vive La France II

出题人: lzz

难度评价: **Easy**

解法:

一个滑膛枪兵等于给波利娜回两点士气，因此我们的策略显然是先打重骑兵->士气不够了再找滑膛枪兵补血。注意到一个重骑兵扣四点士气，简单模拟（找规律）可得，当 $2b - 2 > a$ 时，波利娜可以赢得胜利。

std:

```
#include<stdio.h>
int main(void)
{
    int t,a,b;
    scanf("%d",&t);
    for (int i=1;i<=t;++i)
    {
        scanf("%d%d",&a,&b);
        printf("Case #d: ",i);
        puts((b-1>a/2+a%2)?"Il est temps de se rendre.":"Vive La France!");
    }
    return 0;
}
```

D: 就是这么壕

出题人: ZHR

难度评价: **Hard**

解法:

设第 i 个路灯在位置 X_i 处, 半径为 R_i , 则路灯能照射到的范围为 $[X_i - R_i, X_i + R_i]$ 。

我们要做的就是求出最长的未被覆盖的区间。

考虑贪心加排序。

先求出所有的路灯能照射到的范围为 $[X_i - R_i, X_i + R_i]$, 将它们两个作为一个结构体, 按 $X_i - R_i$ 排序。

然后我们从左到右, 不断更新 $X_i + R_i$ 的最大值。如果某个区间的 $X_i - R_i$ 大于之前 $X_i + R_i$ 的最大值, 那么说明这里有一个未覆盖的区间。

由于路灯半径是一起扩大的, 所以我们只需要求出所有未被覆盖的区间的最大值, 再除以2 (向上取整) 即可得到路灯需要扩大的半径。

记得处理边界, 也就是左右两端的未覆盖区间, 这里不能除以2 (因为只有一端有路灯)。

最后, 当答案为0时, 记得去掉前导零。

这道题的数据质量非常高, 成功干掉了一堆假算法和没考虑边界情况的算法。

我们看到一些同学tle, 这是因为他们多组数据清空时将整个数组给清空了。然而这题有一组数据中 $T = 60000$, 而数据都是小数据 (只有5个路灯), 这么小的数据, 你却清空一个 10^5 级别的大数组6万次, 当然会超时。

有些同学WA了很多发, 其实他们已经通过了 $N = 100000$ 的大数据, 可惜没有通过一些边界的hack情况。我们在附件里提供了这道题目的全部数据, 你可以自己回去对拍。

std:

```
#include<stdio.h>
#include<stdlib.h>
#define maxn 200005
#define max(a,b) (((a)>(b))?(a):(b))
struct cir
{
    int l,r;
}typedef cir;
int comp(const void *p1,const void *p2)
{
    return ((cir *)p1)->l-((cir *)p2)->l;
}
cir ar[maxn];
int main(void)
{
    int T;
```

```

scanf("%d",&T);
while (T--)
{
    int n,l;
    scanf("%d%d",&n,&l);
    for (int i=0;i<n;++i)
        scanf("%d",&ar[i].l);
    for (int i=0;i<n;++i)
    {
        scanf("%d",&ar[i].r);
        ar[i].l-=ar[i].r;
        ar[i].r=ar[i].l+2*ar[i].r;
    }
    qsort(ar,n,sizeof(cir),comp);
    int premax=ar[0].r,maxdis=0;
    for (int i=1;i<n;++i)
    {
        if (ar[i].l>premax)
            maxdis=max(maxdis,ar[i].l-premax);
        premax=max(premax,ar[i].r);
    }
    if (maxdis%2) ++maxdis;
    maxdis/=2;
    maxdis=max(maxdis,l-premax);
    maxdis=max(maxdis,ar[0].l-1);
    if (maxdis==0)
        puts("0");
    else printf("%d0000000\n",maxdis);
}
return 0;
}

```

E: 天下第一

出题人: 691

难度评价: **Hard**

解法: 这道题从后往前倒推比较好想。首先要考虑一件事: (假设自己不是最厉害的) 无论如何你都要贿赂最厉害的人。显然, 和这个最厉害的在决赛碰面是最好的, 因为这个最厉害的人有机会淘汰标号更高的人。

假设这个人标号是 N , 那么这个人从预赛到决赛可以淘汰 $N-1$ 个人, 这里的淘汰指的是间接淘汰。比如第一轮1打2,3打4,第二轮2打4,1,2,3三个人都可以看做是被4号间接淘汰的。不妨假设这些人标号是从 $N/2+1$ 到 $N-1$ 。如果你自己的标号处于这些人中间, 把最厉害的人贿赂了之后终止循环就行了。因为 $N/2$ 之前的人你都打得过, 你一直打到决赛只需要和他们分到一个半区就行。

如果你的标号比 $N/2+1$ 小, 那么半决赛你就会和编号为 $N/2$ 的人打, 贿赂他就行了, 以此类推贿赂 $N/4$, $N/8$...

那么这样策略是最优的吗? 显然不是。我们考虑一件事, 如果贿赂 $N/2$ 的钱比贿赂 $N/2+1$ 的钱多怎么办? 很简单, 让 N 淘汰 $N/2$, 让 $N/2+1$ 成为那个半决赛和自己碰面的人就行了。同理你也可以让 $N/2$, $N/2+1$ 去替代 $N/4$, $N/8$...

因此我们可以在处理完决赛之后，把 $N/2$ 到 $N-1$ 这些人加入到一个队列中，每回合取佣金最小的人去贿赂他。

(为了避免手写堆，这里的数据范围设置在 N^2 也能通过)

注意多组数据的清空问题。

参考程序：

```
#include<stdio.h>
#define MAXN 4100
#define INF 691691691
int bin[MAXN]; //记录每个人所用的佣金
int vis[MAXN]; //记录该名用户是否在队列中
int N,M;
int get_lowest() //从队列中的人找出佣金最低的，并令其出队
{
    int tot=0, mint=INF;
    for(int i=1; i<=N; i++)
        if(vis[i]==1 && bin[i]<mint){tot=i, mint=bin[i];}
    vis[tot]=0;
    return bin[tot];
}
int main()
{
    int T;
    scanf("%d",&T);
    while(T--){
        for(int i=1; i<=N; i++) vis[i]=0; //多组数据及时清空变量
        scanf("%d %d",&N,&M);
        for(int i=1; i<=N; i++){
            scanf("%d",&bin[i]);
            if(i<=M) bin[i]=0; //比自己弱的人直接记作0，方便统计
        }
        int now=N; //目前最强者的标号
        int tot=N-1; //计数器
        int ans=0; //结果

        vis[N]=1; //将第一个强者入队
        while(1){
            int u=now/2; //计算出强者这一轮淘汰的人数
            ans+=get_lowest(); //到队列中最便宜的老哥，令其出队
            if(M>=now/2+1) break; //如果自己处于这个区间里，结束循环
            while(u--){
                vis[tot--]=1; //让这些被淘汰的老哥们入队
            }
            now/=2;
        }
        printf("%d\n",ans);
    }
}
```

```
    }  
    return 0;  
}
```