

2019年厦门大学铃盛杯C语言积分赛热身赛题解

1.Vive La France

出题人: lzz

难度评价: **Very Easy**

解法1:

注意到图片只有4张，而所有可能的排列只有 $4! = 24$ 种。因此，手动枚举1到4的全排列并不断提交，最坏情况下也只需要提交24次。

std:

```
#include <stdio.h>
int main(void)
{
    const char name[5][20]=
    {
        "",
        "Julie Le Brun",
        "Anne Wesley",
        "Cosette Valjean",
        "Marie Antoinette"
    };
    const int ar[]={2,3,1,4}; //手动枚举1到4的全排列，正解是2314
    for (int i=0;i<4;++i)
        puts(name[ar[i]]);
    return 0;
}
```

解法2:

运用你强大的搜索引擎使用能力找到这些图片的原出处。（圣女战旗）

B. wxw的项链

出题人:sy

难度评价: **Easy**

解法:

显而易见的一道约瑟夫问题。先按照循环节建立好链表，然后每隔m个节点打印一个字母并把它删掉即可。

std:

```

#include <stdio.h>
#include <stdlib.h>

typedef struct Node {
    char value;
    struct Node *next;
} Node;

int main(void) {
    const char loop[] = "wxwnb";
    const int loopn = 5;
    int n, m;
    scanf("%d %d", &n, &m);

    Node *root, *p;
    p = root = malloc(sizeof(Node));
    for (int i = 0; i != n; ++i) {
        p->next = malloc(sizeof(Node));
        p = p->next;
        p->value = loop[i%loopn];
    }
    p->next = root->next;

    Node *currp = root;
    for (int i = 0; i != n; ++i) {
        for (int j = 0; j != m-1; ++j) currp = currp->next;
        Node *tempp = currp->next;
        currp->next = tempp->next;
        printf("%c", tempp->value);
        free(tempp);
    }
    return 0;
}

```

另外的，为了降低难度，使得不会链表的同学也能通过这道题目，我们特意降低了数据大小，实际上 $O(N^2)$ 的做法(不删除节点而只是打标记)也可以通过这道题目。但是如果实现效率不高的话有一定风险 TLE。

(以下程序耗时924ms，差76ms超时)

```

#include <stdio.h>
#include <string.h>
#include <stdbool.h>
int main(void)
{
    int n,m,outn=0,i=1,cnt=0;
    scanf("%d%d",&n,&m);
    bool yose[5005]={0};
    char str[10]= {'\0'};
    strcpy(str,"wxwnb");
    while(outn<n)

```

```
{
    i%=(n+1);
    if(i==0)
        ++i;
    if(yose[i])
        ++i;
    else
    {
        ++cnt;
        if(cnt%m==0)
        {
            yose[i]=true;
            cnt=0;
            ++outn;
            int index;
            if(i%5==0)
                index=4;
            else
                index=i%5-1;
            putchar(str[index]);
        }
        ++i;
    }
}
return 0;
}
```

C. 请问您点的是兔子吗？

出题人: lzz

难度评价: **Very Easy**

解法:

简单的字符串题。注意到C语言标准库中的strstr函数，直接调用即可查找到第一个Software_House的位置。然后手动输出或在这个位置加'\0'再输出即可。

另外，读入可能很长很长，虽然理论上1M的空间勉强能放在栈中，但最好还是使用全局数组！

有些人把strlen函数放在循环的判断部分，这样将会严重超时，可以想想为什么。

std:

```
#include <stdio.h>
#include <string.h>
char str[1000050];
int main(void)
{
    scanf("%s",str);
    char *endpos=strstr(str,"Software_House"),*it=str;
    while (it!=endpos)
```

```
    putchar(*it++);  
    return 0;  
}
```

D. 午时已到

出题人: 691

难度评价: **Easy**

解法:

击倒每名敌人所用的子弹数目是 $\text{ceil}(Hi/K)$, (k 为子弹伤害, ceil 为向上取整)

C语言默认向下取整, 一个比较简单的向上取整实现是先判断 Hi 能否被 k 整除, 否则就把答案加1

好多人答案错误却百思不得其解, 交了好多发。我们考虑一个极端数据, 5000个敌人, 每个敌人5000滴血, 子弹伤害为1, 发射间隔为5000, 那么消灭敌人所需要的时间是为 5000^3 , 也就是 $1.25 * 10^{11}$ 秒, 这个数据会超出int的最大表示范围导致溢出, 所以需要使用**long long**数据类型来存储答案!

好多人时间超限却百思不得其解, 交了好多发。有些人用的是减法来计算而不用除法。如果一次次减, 相当于你最坏情况要做 $1.25 * 10^{11}$ 次减法, 这远远超出了一般计算机一秒之内的运算能力! 所以你应该改用除法!

Tips:一定要认真观察数据范围并考虑极端情况, 来评估你的程序!

std:

```
#include <stdio.h>  
int main(void)  
{  
    int T;  
    scanf("%d",&T);  
    while(T--)  
    {  
        int N,M,K;  
        scanf("%d %d %d",&N,&M,&K);  
        int cnt=0;  
        for(int i=1;i<=N;i++)  
        {  
            int u;  
            scanf("%d",&u);  
            cnt+=u/K;  
            if(u%K!=0)++cnt;  
        }  
        printf("%lld\n",(long long)(cnt-1)*M);  
    }  
    return 0;  
}
```

E. 正义的开开

出题人: lrl

难度评价: **Normal**

解法:

注意到三个事实:

1. 和开开同时加入的会员在刚激活的那一天无法给开开发文件（因为他手上没有文件），只能从激活的第二天开始发送文件。而比开开早加入的会员，在开开激活服务器后就可以给开开发送文件。
2. 假如某个会员在第 X 天向开开发送文件，那么这个文件可以从 $X - 1$ 天时从堂主那里得到。
3. 理想情况下向开开发送的文件全部都不重复。（发送的前一天时下载不同的文件即可）

因此我们有答案:

设: T 为开开入会的日子, N 为总服务器数（不包括版主的服务器）, A 为开开那一批入会的服务器数, K 为总补丁数:

$$\textcircled{1} \text{ Ans} = T + 1 \quad (N - A + 1 \geq K)$$

$$\textcircled{2} \text{ Ans} = T + 1 + \lceil (K - (N - A + 1)) / N \rceil \quad (N - A + 1 < K)$$

注: $\lceil x \rceil$ 的含义是 x 向上取整

std:

```
#include <stdio.h>
int main(void)
{
    int n, k, maxx = -1, cnt = 0;
    scanf("%d%d", &n, &k);

    for (int i = 0; i < n; i++)
    {
        int tem;
        scanf("%d", &tem);
        if (maxx < tem)
        {
            maxx = tem;
            cnt = 1;
        }
        else if (maxx == tem)
            cnt++;
    }

    int ans, pre = n + 1 - cnt;
    if (pre >= k)
        ans = maxx + 1;
    else
    {
        ans = maxx + 1 + (k - pre) / n;
    }
}
```

```
        if ((k - pre) % n != 0)
            ans++;
    }

    printf("%d\n", ans);
    return 0;
}
```