

A. 不破不立！我们是冠军！

题解：转义字符，或者单个字符变量保存，都可以打引出双引号，具体看代码

```
#include<stdio.h>

int main()
{
    int n;
    scanf("%d", &n);
    while(n-->0) printf("\nEDG NB\n");
    return 0;
}
```

```
#include<stdio.h>

int main()
{
    int n;
    scanf("%d", &n);
    char ch = ' ';
    while(n-->0)
    {
        printf("%c", ch);
        printf("EDG NB");
        printf("%c\n", ch);
    }
    return 0;
}
```

// char (34) c++可以过，c 编译错误

```
#include<stdio.h>

int main()
{
    int n;
    scanf("%d", &n);
    while(n-->0)
    {
        printf("%c", char(34));
        printf("EDG NB");
    }
}
```

```
        printf("%c\n", char(34));
    }
    return 0;
}
```

B. 小豪的室友

题解：最终状态有两种：

要么 wbwbwbw...

要么 bwbwbwb...

所以我们枚举两种情况，对于任何一种情况，分别用 x 和 y 表示没有放对位置的黑书和白书的数量。对于每种情况，我们都先找出最小的交换次数，即 $\min(x, y)$ ，剩下的我们直接从小豪那里拿书替代。

故每种情况最少的操作次数为：

$$\min(x, y) + [\max(x, y) - \min(x, y)] = \max(x, y);$$

则最终结果：

$$\text{ans} = \min(\max(x_1, y_1), \max(x_2, y_2));$$

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

typedef long long ll;
char S[100010];
int max(int x, int y)
{
    if (x > y) return x;
    return y;
}
int min(int x, int y)
{
    if (x > y) return y;
    return x;
}
```

```

int main() {
    int n, len;
    scanf("%d", &n);
    scanf("%s", S);
    int flag1, flag2;
    int sum1, sum2;
    flag1 = flag2 = sum1 = sum2 = 0;
    len = strlen(S);
    for (int i = 0; i < len; i++) {
        if (S[i] == 'w') {
            if (i & 1) flag1++;
            else flag2++;
        }
        else {
            if (i & 1) sum1++;
            else sum2++;
        }
    }
    int ans = min(max(flag1, sum2), max(sum1, flag2));
    printf("%d", ans);
    return 0;
}

```

C. 我爱数论

题解：G(x) 函数其实就是质数的改版，只是 x 为 1 时， $G(1) = 1$ ，这也是一个小小的坑点。

//质数定义：质数又称素数。一个大于 1 的自然数，除了 1 和它自身外，不能被其他自然数整除的数叫做质数；否则称为合数（规定 1 既不是质数也不是合数）。

思路：写个判断质数的函数，循环判断该区间即可。

```

#include<stdio.h>
#include<math.h>

int judge(int x)
{
    if (x == 1) return 1; //坑点
    int q = sqrt(x);

```

```

    for (int i = 2; i <= q; i++)
        if (x % i == 0) return 0;
    return 1;
}

int main()
{
    int l, r;
    scanf("%d %d", &l, &r);

    int ans = 0;
    for (int i = l; i <= r; i++)
        ans += judge(i);

    printf("%d\n", ans);

    return 0;
}

```

D. 聚众打 codeforce

```

#include<stdio.h>

char s[100];
int n;

int main()
{
    scanf("%s %d", s, &n);
    if(s[n-1] == 'o') printf("Yes");
    else printf("No");
    return 0;
}

```

E. szg 的情敌

题解：

暴力解法（小数据）：对于每次读入的两个值 x 和 y ，在区间 $[x, y)$ 上对 $a[i]$ 加一，如果

$a[i]$ 表示在 i 时刻 zyz 面对多少女粉丝。如果 $a[i] > m$ ，则 zyz 会沦

陷。

差分解法（大数据）：请参考泽佬的博客



https://github.com/Chivas-Regal/ZZULI_ACM_PRACTICE/blob/main/Practice/21%E7%BA%A7%E6%AF%8F%E6%97%A5%E4%B8%80%E7%BB%83.md#%9%E6%9C%8817%E6%97%A5%E6%80%9D%E7%BB%B4%E6%A8%A1%E6%8B%9F%E6%9E%84%E9%80%A06%E5%89%8D%E9%9D%A2%E7%AE%97%E6%B3%95%E5%B7%A9%E5%9B%BA

```
#include <stdio.h>
#include <string.h>
//感谢 szg 友情提供。
typedef long long ll;
int max(int x, int y)
{
    if (x > y) return x;
    return y;
}
int a[200010];
int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        int n, m;
        scanf("%d %d", &n, &m);
        int maxx = -1;
        int flag = 0;
        int sum = 0;
        memset(a, 0, sizeof a); // 将数组初始化的函数，将数组初始化为 0
        for (int i = 0; i < n; i++) {
            int x, y;
            scanf("%d %d", &x, &y);
            maxx = max(maxx, y);
            a[x]++;
            a[y]--;
        }
        for (int i = 0; i <= maxx; i++) {
            sum += a[i];
        }
    }
}
```

```

        if (sum > m) {
            flag = 1;
            break;
        }
    }
    if (!flag) puts("NO");
    else puts("YES");
}
}

```

F. 这真的是签到题啊

题解：

基本数学知识，坑点在于最后的结果会爆 int，可以开 double 或者 long long 存

```

#include<stdio.h>

int main()
{
    int T, n;
    scanf("%d", &T);
    while(T--)
    {
        scanf("%d", &n);
        // printf("%lld\n", (long long) n*(n+1)/2);
        printf("%.0lf\n", (double) n*(n+1)/2);
    }
    return 0;
}

```

G. 圣诞树

思路：打印图形嘛，花点时间其实可以做出来的。

可以写两个函数：（1）打印三角形函数（2）打印矩形函数。

这样便方便很多了。

这题一半都是格式错误，这里有一个小小的坑，就是你得判断空格的

数量，也就是要判断“圣诞树”的哪一部分最宽。明显，最宽的是第三部分的三角形的宽度或者是土壤的宽度。

```
#include<stdio.h>

int max(int x, int y)
{
    return x > y ? x : y;
}

//输出三角形
void print1(int pre, int r)
{
    //pre 为该图形前面的空格数量, r 为图形半径
    for (int i = 1; i <= r; i++)
    {
        int num = pre + r - i; //空格数量
        for (int j = 1; j <= num; j++) printf(" ");
        for (int j = 1; j <= i * 2 - 1; j++) printf("*");
        printf("\n");
    }
}

//输出矩形
void print2(int pre, int r, int h)
{
    for (int i = 1; i <= h; i++)
    {
        for (int j = 1; j <= pre; j++) printf(" ");
        for (int j = 1; j <= r * 2 - 1; j++) printf("*");
        printf("\n");
    }
}

int main()
{
    int r, d;
    int r1, h1, r2, h2;
    scanf("%d%d", &r, &d);
    scanf("%d%d%d%d", &r1, &h1, &r2, &h2);

    //宽度转换为半径
    r = r / 2 + 1;
```

```

    r1 = r1 / 2 + 1;
    r2 = r2 / 2 + 1;

    //圣诞树的对称轴到最前面的宽度
    int maxn = max(r + d, r2);

    //三个三角形
    print1(maxn - r, r);
    print1(maxn - r - d / 2, r + d / 2);
    print1(maxn - r - d, r + d);

    //两个矩形
    print2(maxn - r1, r1, h1);
    print2(maxn - r2, r2, h2);

    return 0;
}

```

H. 快来 AC 这道题

题解：矩阵排列规律很容易看出，具体实现思路为找出左上到右下这个对角线的数字之间的关系，再确定这个对角线上的元素对应的行和列的元素，对角线上元素的递推关系为 $a[i][i] = a[i-1][i-1] + 2*i$

```

#include<stdio.h>

int n;
int a[110][110];

int main()
{
    scanf("%d", &n);
    a[0][0] = 1;
    for(int i = 1; i < n; i++) a[i][i] = a[i-1][i-1] + 2*i;
    for(int i = 1; i < n; i++)
    {
        for(int j = i-1; j >= 0; j--)
        {
            a[i][j] = a[i][j+1] + 1;
            a[j][i] = a[j+1][i] - 1;
        }
    }
}

```



```

    }
    for(int i = 0; i < n; i++)
    {
        for(int j = 0; j < n; j++)
        {
            printf("%d ", a[i][j]);
        }
        printf("\n");
    }
    return 0;
}

```

1. 小轩的梦

思路 1:

$$\text{Ans} = n * n * (n - 1) * (n - 1) * (n - 2) * (n - 2) * (n - 3) * (n - 3) * (n - 4) * (n - 4) / A_5^5$$

第一次有 $n * n$ 种选择，因为不能在同行，也不能在同一列，所以第一次选择后，所在的行和所在的列删掉即可。删掉后便成为和删掉前一样的子问题，即有 $(n - 1) * (n - 1)$ 种选择，后面以此类推。

但是，注意，因为是 5 个相同的彩灯，所以需要除掉 A_5^5 。

思路 2:

$$\text{Ans} = C_n^5 * A_n^5 = A_n^5 * A_n^5 / A_5^5$$

首先从 n 行中任选 5 行，也即 C_n^5 。然后对于这选出的五行，因为不能在同一列，第一行有 n 种选择，第二行有 $n - 1$ 种选择。。。以此类推。

最后：说一下坑点，这题若运算顺序不恰当，会爆 long long int，所以，假设： $\text{sum} = n * (n - 1) * (n - 2) * (n - 3) * (n - 4)$

则, $\text{Ans} = \text{sum} / A_5^5 * \text{sum}$ 。提一下: 这题若用 `double` 的话会精度丢失, 所以会被判错。

```
#include<stdio.h>
#define ll long long

int main()
{
    ll n;
    scanf("%lld", &n);

    ll sum = 1;
    for (int i = 0; i < 5; i++)
        sum *= n - i;

    if (n < 5) printf("0");
    else printf("%lld", sum / 120 * sum);

    return 0;
}
```

J. 小轩的好数 VS 小孟的真~好数

题解:

本题考察函数的简洁性, 对于输入的 n , 从 1 枚举到 \sqrt{n} , 对于这个区间的每个 i , 如果 i 能被 n 整除, 则对 i 和 n / i 进行判断。暴力枚举即可。

注意: 1. 一定要注意 答案的输出, 一定要和题目中的一样, 如 **N0** 不是 **N0**, 是数字 0 (正确率对味儿了)

2. 从 $1 \sim n$ 多了没必要的枚举, 时间复杂度很高, 枚举到 \sqrt{n} 就行

3. 枚举区间时, `for` 循环的判断条件 应该是 $i * i \leq n$, 如果是 $i \leq n / i$; 则没有考虑 n / i 会取整, 而导致答案错误。

```

#include <stdio.h>

typedef long long ll;
bool check(ll i) {
    while (i) {
        ll x = i % 10;
        if (x != 4 && x != 7) return false;
        i /= 10;
    }
    return true;
}

int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        ll n;
        scanf("%lld", &n);
        int flag = 0;
        for (ll i = 1; i * i <= n; ++i) {
            if (n % i == 0) {
                if (check(i)) {
                    puts("YES");
                    flag = 1;
                    break;
                }
                if (check(n / i)) {
                    puts("YES");
                    flag = 1;
                    break;
                }
            }
        }
        if (!flag) puts("NO");
    }
}

```

K. 小轩的衣服

题解：（防 ak 题）

首先如果 $s == k$ ，则不管怎么分，都能找到 n 个连续的盒子，使得

这些盒子中的衣服的总量为 k 。

之后考虑 $s < k$ ，则不管怎么分，都不能找到 n 个连续的盒子，使得这些盒子中的衣服的总量为 k 。

剩下我们考虑 $s > k$ ，考虑命题，对于给定的 s, n, k ，每个分配方案是否都可以找到若干个连续的盒子，使得这些盒子中的衣服的总量为 k 。

可以将命题转换为，对于给定的 s, n, k ，每个分配方案是否都找不到若干个连续的盒子。

则我们可以考虑一个有 n 项的序列，序列的总和为 s ，则题意可以转换为对于任意区间 $[L, R]$ ，该区间的和不为 k 。

则对于前 k 项，将前 $k - 1$ 项都设为 1，第 k 项设为 $k + 1$ ；

对于第 $(k + 1) \sim 2k$ 项，将第 $(k + 1)$ 项~第 $(2k - 1)$ 项设为 1，第 $2k$ 项设为 $k + 1$ 。

以此类推。则不管怎么找区间，区间的和都不为 k 。

该序列的 总和 $sum = (n / k) * 2k + n \% k$ ，其中 n / k 向下取整， $n \% k$ 即 n 对 k 取余。

如果 $s < sum$ 则构不成该序列，所以每种分配方案，都能找到 n 个连续的盒子，使得这些盒子中的衣服的总量为 k 。

如果 $s \geq sum$ 则可以构成该序列，所以存在一种分配方案，找不到 n 个连续的盒子，使得这些盒子中的衣服总量为 k 。

```
#include <stdio.h>

typedef long long ll;
```

```
int main()
{
    int t;
    scanf("%d", &t);
    while (t--)
    {
        int s, n, k;
        scanf("%lld %lld %lld", &s, &n, &k);
        if (s == k)
        {
            puts("YES");
        }
        else
        {
            int sum = n / k * 2 * k + n % k;
            if (sum <= s) puts("NO");
            else puts("YES");
        }
    }
}
```