**1.我要 AK 新生赛**

新生赛是考验大一新生在入学以来所掌握的编程能力，$AK$是每个选手在比赛中的最大追求。$AK$ :指在一次比赛中把全部题目都写出来了。所以如果你想$AK$新生赛，那就输出“wo yao ak xin sheng sai”吧!

**输入**

无

**输出**

wo yao ak xin sheng sai

# Code

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    cout<<"wo yao ak xin sheng sai";
    return 0;
}
```

**2.硬币**

Luo Weijie同学正在被一个硬币游戏困扰，于是他去寻找他的好朋友Lin Chao帮忙。

硬币有字面和花面，Luo Weijie面前存在$m$个字面朝上的硬币，在一步里,Luo Weijie可以将至多$n$个硬币翻面，Luo Weijie想知道他能不能通过若干步操作，使面前的所有硬币都变为花面朝上的状态(即不存在字面朝上)。如果能的话，Lin Chao会告诉他至少需要多少步。

**输入**

输入仅一行，包括两个整数 $m, n(0 \leq m, n \leq 10^{12})$。

**输出**

如果Luo Weijie能做到将面前的所有硬币都变为花面朝上的状态，则在一行一个整数，表示Lin Chao的答案，否则在一行输出 $error$ ，表示不能做到。

**示例1**

**输入**

8 6

**输出**

2

# Code

```cpp
#include <bits/stdc++.h>
using namespace std;

int main()
{
    long long  m,n;
    cin>>m>>n;
    if(m==0){
        cout << 0;
        return 0;
    }
    if (n==0){
        cout << "error";
        return 0;
    }
    long long  ans = ceil((m*1.0)/(n*1.0));
    cout << ans << endl;
    return 0;
}
```

**3.矩阵**

今天线性代数课上，老师教了矩阵相乘的知识点。但是小林同学因为昨晚熬夜打代码，结果课上睡着了。现在他向上过课的你请教，你能帮助一下他嘛。他给你两个矩阵$A$和$B$，让你帮他计算一下这两个矩阵的乘积，需要注意的是，要规模匹配的矩阵才可以相乘。即若$A$有$R_a$行、$C_a$列，$B$有$R_b$行、$C_b$列,则只有$C_a$与$R_b$相等时，两个矩阵才可以相乘。

**输入**

先后输入两个矩阵$A$和$B$。对于每个矩阵，首先一行中给出其行数$R$和列数$C$，随后$R$行，每行给出$C$个整数。

**输出**

如果两个矩阵的规模是匹配的，则按照输入的格式输出乘积矩阵$AB$，否则输出$Error$。

**示例1**

**输入**

2 3
1 2 3
4 5 6
3 4
7 8 9 0
-1 -2 -3 -4
5 6 7 8

**输出**

2 4

20 22 24 16
53 58 63 28

**示例2**

**输入**

3 2
38 26
43 -5
0 17
3 2
-11 57
99 68
81 72

**输出**

Error

**数据范围**

输入保证$0 < R, C < 10$，并且所有矩阵内整数的绝对值不超过100。

# Code

```
#include <bits/stdc++.h>
using namespace std;
int a[200][200];
int b[200][200];
int main()
{
    int c1, c2, c3, c4;
    cin >> c1 >> c2;
    for (int i = 1; i <= c1; i++)
    {
        for (int j = 1; j <= c2; j++)
        {
            cin >> a[i][j];
        }
    }
    cin >> c3 >> c4;
    for (int i = 1; i <= c3; i++)
    {
        for (int j = 1; j <= c4; j++)
        {
            cin >> b[i][j];
        }
    }
```

```
    if (c2 != c3)
    {
        printf("Error");
    }
    else
    {
        cout << c1 << " " << c4 << endl;
        for (int i = 1; i <= c1; i++)
        {
            for (int j = 1; j <= c4; j++)
            {
                int sum = 0;
                for (int k = 1; k <= c2; k++)
                    sum += a[i][k] * b[k][j];

                if (j != c4)
                    cout << sum << " ";
                else
                    cout << sum << endl;
            }
        }
    }
}
```

**4.求因子**

为了让新生都能$AK$新生赛，林学长花了两年半的时间出了这题。林学长现在给你一个数$n$,让你求出这个数所有的正因子。

**输入**

输入一个正整数 $n(1 \leq n \leq 10^{12})$。

**输出**

输出两行，一行表示$n$的正因子的个数，另一行输出$n$的所有正因子（按照从大到小的顺序输出并用空格隔开）。

**示例1**

**输入**

6

**输出**

4

6 3 2 1

**示例2**

**输入**

5

**输出**

2

5 1

# Code

```
#include <bits/stdc++.h>
using namespace std;
long long s[1000000];
int main()
{

    long long n;
    cin >> n;
    int j = 0;
    for (long long i = 1; i * i <= n; i++)
    {
        if (n % i == 0)
        {
            s[++j] = i;
            if (i != n / i)
                s[++j] = n / i;
        }
    }
    sort(s + 1, s + 1 + j);
    cout << j << "\n";
    for (int i = j; i >= 1; i--)
        cout << s[i] << " ";
}
```

**5.01字符串**

学习部部长突然叫住小民同学，想考小民一个问题，给出一个长度为$n$的只含0和1的字符串，如果字符串中每段连续的'1'或者连续的'0'的个数都为偶数，那这个字符串是完美的，反之如果每段连续的'0'或者'1'不是偶数，那这个字符串就是不完美的。

本题有多组数据，对于每组数据，输出一个将不好的0,1串变成好的所需要的最少操作次数。

对于每次操作，可以将0,1串中任意的'0'改成'1'，或者将'1'修改成'0'。

**注意**

本题数据比较大，建议使用scanf输入，printf输出！

怕你们不记得，来自学习部部长的馈赠。

int n;

char s[10000];

scanf("%d",&n)，scanf("%s",&s)，printf("%d",n);

**输入**

第一行输入一个正整数 $T$（$T \leq 10000$），表示数据组数。

对于每组数据，
第一行为一个偶数 $n(2 \leq n \leq 2 * 10^5)$，表示 0,1 串的长度。
第二行为一个长度为n的01串。

**输出**
每行输出一个数，表示将不好的0,1串变成好的所需要的最少操作次数。

**示例1**

**输入**
5
10
1110011000
8
11001111
2
00
2
11
6
100110

**输出**
3
0
0
0
3

**样例说明**
S为字符串，s1表示字符串的第一个字符
对于样例1：
S=1110011000
可以把s4改成1，s6改成0，s7改成0，至少要3次操作才能使S变成完美。
对于样例2：
S=11001111
已经是完美的了，不需要操作。

# Code

```cpp
#include <iostream>
using namespace std;
int t, n;
int cnt = 0;
char a[300010];
int main()
{
    cin >> t;
    while (t--)
    {
        scanf("%d", &n);
        cnt = 0;
        scanf("%s", &a);
        for (int i = 1; i < n; i += 2)
        {
            if (a[i] != a[i - 1])
            {
                a[i - 1] = a[i];
                cnt += 1;
            }
        }
        printf("%d\n", cnt);
    }
    return 0;
}
```

**6.笑脸字符串**

小民给自己取个网名，他想取一个只包含字符 `_` 和 `^` 的网名，并满足一定条件，即只有形如 `^_^` 和 `^^` 的连续子串可以出现在该名字中，且这些子串能够覆盖整个名字，不同子串间可以重叠。每次操作可以在名字中插入一个字符 `_` 或一个字符 `^`，求最少需要多少次操作才能使其符合要求。

**输入**

本题有$T$组数据，对于每组数据只有一行输入，输入一个只包含“`_`”和“`^`”的字符串。

**输出**

每组数据有一行输出，输出将网名修正的最少操作次数。

**示例1**

**输入**

7

```
^_____^
___^_^^^_^___^
^_
^
^_^^^^^_^_^^
___^^
_
```

**输出**

5
5
1
1
0
3
2

**样例说明**

对于样例1：
最少花费五次操作，即修改成 `^_^_^_^_^_^_^` 。
对于样例3：
只需要一次操作，就可将其修改成 `^_^` 。
对于样例4：
只需要一次操作，就可修正，即 `^^` 。
对于样例5：
已经是正确的名字，不需要操作，所以是0次操作。

# Code

```
#include <bits/stdc++.h>
using namespace std;

int T;
string x;

int main()
{

    cin >> T;
    while (T--)
    {
        cin >> x;
        char pre;
        long long ans = 0;
```

```
        for (int i = 0; i < x.size(); i++)
        {
            if (i == 0)
            {
                pre = x[i];
                if (x[i] == '_')
                    ans++;
            }
            else if (x[i] == '_')
            {
                ans++;
                pre = '_';
            }

            else
            {

                if (pre == '_')
                {
                    ans--;
                    pre = '^';
                }
            }
        }

        if (x.size() == 1 && x[0] == '^')
        {
            cout << 1 << endl;
            continue;
        }

        if (x[0] == '_')
            ans++;
        cout << ans << endl;
    }
    return 0;
}
```

**7.程序员**

小民在参加评选最佳打码选手的会议，一共有$n$个评委，$m$个选手，每轮投票每个评委都会在剩下的选手中选一个人。在第一轮投票前，有$m$个选手可以选择；每轮投票后，只保留有最多票数的选手；当只剩下一个选手时，评选结束。

请判断无论怎样投票评选都会结束吗?

**输入**

本题有$T$个数据，每行输入俩个数$n$，$m(1 \leq n, m \leq 10^6)$;

**输出**

如果无论如何都会结束，那就输出“YES”，否则输出‘NO’。

**示例1**

**输入**
5
3 2
4 2
5 3
1000000 1000000
1 1000000

**输出**
YES
NO
YES
NO
YES

**样例说明**
121 表示 1号评委选择选手1,2号评委选择选手2,3号评委选择选手1。以此类推
对于样例1：
第一轮投票中，投票情况可能有 111,112,121,122,211,212,221,222，无论如何都会剩下一个人。
对于样例2：
每轮投票中，俩名评委选择选手1，俩名评委选择选手2，那样每轮投票都是持平，那评选就无法结束。所以输出“NO“。

# Code

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int t;
    scanf("%d", &t);
```

```
    while (t--)
    {
        int flag = 0;
        int n, m;
        scanf("%d%d", &n, &m);
        if (n <= m && n != 1)
        {
            printf("NO\n");
            continue;
        }

        for (int i = 2; i * i <= n; i++)
        {
            if (n % i == 0 && i <= m)
            {
                printf("NO\n");
                flag = 1;
                break;
            }
        }
        if (!flag)
            printf("YES\n");
    }
    return 0;
}
```