

## Contents

1. Cơ sở.....	1
1. Hoán vị kế tiếp.....	1
2. Penny game.....	3
4. Số nén tối giản.....	4
5. Thi Đấu.....	6
6. Đồ xe tối ưu.....	7
7. Trò chơi dò mìn.....	8
8. Phân tích số nguyên.....	10
9. Rick Flag.....	11
10. Bat man.....	12
11. Captain Boomerang.....	14
12. Next round.....	15
13 Xây tháp.....	16
14 Sưu tầm tem.....	17
15 Tín hiệu mật.....	18
16 Bóng đá.....	19
17. Chữ số n.....	20
18. Xúc xắc.....	21
19. Xếp que.....	22
20. Card game.....	24
21. Bàn cờ.....	25
22. Cát giấy.....	27
23. Kim Cương.....	28
24. Cặp số đẹp.....	29
25. Lựa chọn dãy số.....	30
25. Tìm lại dãy số.....	31
26. Chỉ là sắp xếp.....	32
27. Chuyển đổi.....	34
28. Ký túc xá.....	35
29. Trai hay gái.....	36

30.	Sắp xếp.....	37
31.	Pangram.....	38
32.	Giờ học lý thú.....	39
33.	Ký tự giống nhau.....	40
34.	Số nguyên tố kế tiếp.....	41
35.	XOR.....	43
36.	Đong gạo.....	43
37.	Trang trí bàn cờ.....	44
38.	Chiếc vé may mắn.....	46
39.	Hoán vị kế tiếp.....	47
40.	Giá trị của năm.....	49
41.	Đếm giày.....	50
42.	Ghép hình.....	51
43.	Lặp lại các kí tự.....	52
	PTIT1211 - Lặp lại các kí tự.....	52
44.	Giai thừa 2.....	53
45.	Đếm số lần lặp.....	54
46.	Tráo bài.....	56
47.	Chuyển đổi hệ cơ số(Bài dễ).....	57
48.	Chỗ ngồi.....	58
49.	Cột có tích lớn.....	59
50.	Bỏ phiếu.....	61
51.	Tổ chức kì thi.....	62
52.	Bán vé số.....	63
53.	Biến đổi chuỗi.....	65
54.	Find the cow:.....	65
55.	Xóa bit.....	67
56.	Tổng bình phương.....	68
57.	Đỗ xe.....	69
58.	Blackjack.....	70
59.	Nhìn và đọc.....	71
60.	Tính lãi suất.....	72
61.	BIT operator.....	73

62.	Tuần lễ công dân.....	74
63.	Test IQ.....	76
64.	Đàn kiến.....	78
65.	Chia Kẹo.....	79
66.	Lũy Thừa.....	81
67.	Tung đồng xu.....	82
68.	Đoạn thẳng lớn nhất.....	83
69.	Modulo.....	85
70.	Quán café.....	87
71.	Bút màu.....	88
72.	Luyện tập ACM.....	89
73.	Hình hộp chữ nhật.....	90
74.	Pha nước cam.....	91
75.	Điểm cân bằng.....	92
76.	Hai xâu có các kí tự giống nhau.....	93
77.	Dãy số Hailstone.....	95
78.	Xếp hạng.....	96
79.	Các cặp giai thừa.....	98
80.	Số ma thuật.....	100
81.	Chữ số cuối cùng.....	101
82.	Min Max.....	102
83.	Hoán vị.....	104
84.	Quyết chiến.....	105
85.	Chẵn lẻ.....	106
86.	Mã hóa xâu.....	107
87.	Về quê.....	109
88.	Cùng bet với EartchShaker.....	111
89.	Điện tử số.....	112
90.	Tặng quà.....	113
91.	Tiệc bia.....	114
92.	Phép dịch.....	115
93.	Số may mắn.....	117
94.	Tìm chữ số còn thiếu.....	118

95.	Chọn chỗ.....	122
96.	Bữa tiệc sinh nhật.....	123
97.	Vi khuẩn.....	124
98.	Alex gửi tiền ngân hàng.....	125
99.	Cơ số 3.....	125
100.	Kim tự tháp.....	127
101.	Số nguyên.....	128
102.	Thăng chức.....	130
103.	Khóa vòng.....	132
104.	Cấp số cộng hay cấp số nhân.....	133
105.	ABC.....	136
106.	Phương án khuyến mại.....	137
107.	Dãy con tăng dần tự nhiên bậc K.....	139
108.	Số may mắn.....	140
109.	ĐOán số.....	141
110.	Ptit sumer.....	143
111.	Mã hóa thư.....	144

## 1. Cơ số

Cho 1 số ở dạng nhị phân. Chuyển số đó sang dạng bát phân.

Input

Một dòng gồm 1 số ở dạng nhị phân (Không quá 63 chữ số).

Output

Đáp án của bài toán.

Example

Test 1:

Input:

1100

Output:

14

Test 2:

Input:

11001100

Output:

314

### 1. Hoán vị kế tiếp

Trong bài này, bạn hãy viết chương trình nhận vào một chuỗi (có thể khá dài) các ký tự số và đưa ra màn hình hoán vị kế tiếp của các ký tự số đó (với ý nghĩa là hoán vị có giá trị lớn hơn tiếp theo nếu ta coi chuỗi đó là một giá trị số nguyên).

Chú ý: Các ký tự số trong dãy có thể trùng nhau.

Ví dụ:

123 -> 132

279134399742 -> 279134423799

Cũng có trường hợp sẽ không thể có hoán vị kế tiếp. Ví dụ như khi đầu vào là chuỗi 987.

#### Dữ liệu vào

Dòng đầu tiên ghi số nguyên  $t$  là số bộ test ( $1 \leq t \leq 1000$ ). Mỗi bộ test có một dòng, đầu tiên là số thứ tự bộ test, một dấu cách, sau đó là chuỗi các ký tự số, tối đa 80 phần tử.

#### Dữ liệu ra

Với mỗi bộ test hãy đưa ra một dòng gồm thứ tự bộ test, một dấu cách, tiếp theo đó là hoán vị kế tiếp hoặc chuỗi "BIGGEST" nếu không có hoán vị kế tiếp.

Example

**Input :**

3

1 123

2 279134399742

3 987

**Output :**

1 132

## 3 BIGGEST

```

1. using namespace std;
2. void xuly(char s[]){
3.     int x[100],n = strlen(s);
4.     for(int j=0;j<n;j++)
5.         x[j] = (int) (s[j]-'0');
6.     int i = n-2;
7.     while(i>=0&& x[i]>=x[i+1]) i--;
8.     if(i<0)
9.         cout<<"BIGGEST\n";
10.    else{
11.        int k = n-1;
12.        while(x[i]>=x[k]) k--;
13.        swap(x[i],x[k]);
14.        int r = i+1,s = n-1;
15.        while(r<s){
16.            swap(x[r],x[s]);
17.            r++;s--;
18.        }
19.        for(int j=0;j<n;j++)
20.            cout<<x[j];
21.        cout<<"\n";
22.    }
23. }
24. main(){
25.     int test;
26.     cin>>test;
27.     while(test--){
28.         int stt;
29.         cin>>stt;
30.         char s[100];
31.         cin>>s;
32.         cout<<stt<<" ";
33.         xuly(s);
34.     }
35. }

```

## 2. Penny game

Penny Game là một trò chơi đơn giản cho hai người chơi. Trò chơi này yêu cầu mỗi người chơi chọn một dãy duy nhất ba mặt đồng xu ví dụ như **HEADS TAILS HEADS (HTH)**. Các đồng xu sẽ được tung liên tiếp cho đến khi xuất hiện một trong hai dãy đã được chọn lúc đầu. Người chơi nào chọn được dãy xuất hiện đầu tiên sẽ là người thắng cuộc.

Bạn hãy viết một chương trình đọc một chuỗi 40 giá trị tung đồng xu liên tiếp và xác định mỗi một chuỗi ba mặt đồng xu xuất hiện bao nhiêu lần. Hiển nhiên, ta sẽ có 8 chuỗi 3 mặt đồng xu lần lượt là: TTT, TTH, THT, THH, HTT, HTH, HHT và HHH. Các chuỗi này có thể xếp chồng lên nhau. Ví dụ ta có 40 xu liên tiếp ở mặt HEADS thì sẽ có 38 chuỗi HHH xuất hiện

**Dữ liệu:** Dòng đầu tiên ghi số  $1 \leq P \leq 1000$  là tổng số bộ test. Mỗi bộ test gồm 2 dòng:

- Dòng đầu tiên chứa số N là thứ tự bộ test.
- Dòng thứ hai là dãy 40 ký tự mô tả chuỗi giá trị tung đồng xu trong đó chỉ gồm hai ký tự là H và T. Không có bất cứ dấu cách nào trong các dòng dữ liệu vào.

**Kết quả:** Ghi ra màn hình một dòng cho mỗi bộ test trong số đầu tiên là thứ tự bộ test. Tiếp theo là 8 số cho biết số lần xuất hiện của mỗi chuỗi 3 đồng xu theo thứ tự ở trên. Mỗi số viết cách nhau một dấu cách. Sẽ có 9 dấu cách trong mỗi dòng kết quả.

### Example

**Input:**

4

1

HH

2

TT

3

HHTTTTHHTTTHHTHTHTHTTTTHHHTHTTHTTTHTTTHTH

4

HTHTHHHTHHHTHTHHHHHTTTHTTTTTHTTTTHTHHHHHT

### Output:

1 0 0 0 0 0 0 0 38

2 38 0 0 0 0 0 0 0

3 4 7 6 4 7 4 5 1

4 6 3 4 5 3 6 5 6

```

3. void xuly(string s){
4.     int so[10];
5.     for(int i=1;i<9;i++)
6.         so[i] = 0;
7.     int n = s.length();
8.     for(int i=0;i<n-2;i++){
9.         string temp;
10.        for(int j=i;j<i+3;j++)
11.            temp = temp + s[j];
12.        if(temp == "TTT") so[1]++;
13.        if(temp == "TTH") so[2]++;
14.        if(temp == "THT") so[3]++;
15.        if(temp == "THH") so[4]++;

```

```

16.             if(temp == "HTT") so[5]++;
17.             if(temp == "HTH") so[6]++;
18.             if(temp == "HHT") so[7]++;
19.             if(temp == "HHH") so[8]++;
20.         }
21.         for(int i=1;i<9;i++)
22.             cout<<so[i]<<" ";
23.     }
24.     main(){
25.         int test;
26.         cin>>test;
27.         while(test--){
28.             int stt;
29.             cin>>stt;
30.             string s;
31.             cin>>s;
32.             cout<<stt<<" ";
33.             xuly(s);
34.         }
35.     }

```

#### 4. Số nén tối giản

Ta gọi phép nén một số nguyên là tính tổng các chữ số của nó. Dễ thấy, sau một số phép nén, thì số còn lại chỉ có một chữ số và ko nén được nữa. Ta gọi số đó là số nén tối giản.

Ví dụ cho số 86. Sau phép nén thứ nhất ta đk:  $8+6=14$ . Sau phép nén thứ 2:  $1+4=5 \Rightarrow$  Số nén tối giản của 86 là 5.

Cho một số nguyên hãy tìm số nén tối giản của nó.

Input

Dòng đầu chứa số bộ test.

Mỗi dòng tiếp theo chứa 1 bộ test gồm 1 số nguyên dương. ( $\leq 10^9$ )

Mỗi dòng

Output

Mỗi dòng một số nén tối giản tương ứng.

Example

**Input :**

**3**

**43**

**111**

**57871**

**Output :**



7

3

1

```

1. int nen(long n){
2.     while(n>9){
3.         long i=n;
4.         int tong = 0;
5.         while(i>0){
6.             tong += i%10;
7.             i /= 10;
8.         }
9.         n = tong;
10.    }
11.    return n;
12. }
13. main(){
14.     long test;
15.     cin>>test;
16.     while(test--){
17.         long n;
18.         cin>>n;
19.         cout<<nen(n)<<"\n";
20.     }
21. }
22.

```

### 5. Thi Đấu

Trong cuộc thi ACM, các đội thi đấu với nhau. Thứ hạng của mỗi đội phụ thuộc vào: số lượng bài giải được và số điểm bị phạt (điểm phạt phụ thuộc vào thời gian và số lần nộp bài trước khi bài đó được giải đúng). Điểm phạt bằng thời gian nộp bài đúng cộng với 20 phút cho mỗi lần nộp sai. Nếu một bài không giải được thì những lần nộp sai bài đó không bị tính điểm phạt. Như vậy nếu đội giải đúng vào lần nộp thứ 2, sau 20 phút thì điểm phạt sẽ là 40 điểm. Đội thắng là đội giải được nhiều nhất, nếu có nhiều đội có số bài giải giống nhau thì đội bị phạt ít hơn sẽ giành chiến thắng.

**Input:** Cuộc thi ACM lần này chỉ có 4 bài.

Dòng 1: số n tương ứng với số đội thi

Dòng 2 đến dòng n+1: Mỗi dòng ghi

<Tên> <p1Sub> <p1Time> <p2Sub> <p2Time>....<p4Time>

Thành phần đầu tiên là tên đội, không chứa dấu cách. Sau đó là thông tin về đội đó: p1Sub số lần đội nộp lời giải cho bài 1, p1Time là thời điểm đội nộp đúng, nếu đội không giải đúng được bài 1 thì p1Time=0, tương tự với 3 bài còn lại.

**Output:** Ghi tên đội thắng, số bài giải được, số điểm phạt

**Ví dụ cho Input và Output:**

INPUT	OUTPUT
4 Stars 2 20 5 0 4 190 3 220 Rockets 5 180 1 0 2 0 3 100 Penguins 1 15 3 120 1 300 4 0 Marsupials 9 0 3 100 2 220 3 80	Penguins 3 475

```

1. struct doi{
2.     char ten[100];
3.     int sobai, phat, nop[5], time[5];
4. };
5. main(){
6.     doi ds[101];
7.     int n; cin>>n;
8.     for(int i = 1; i <= n; i++){
9.         cin>>ds[i].ten;
10.        for(int j = 1; j <= 4; j++){
11.            cin>>ds[i].nop[j]>>ds[i].time[j];
12.            if(ds[i].time[j] != 0){
13.                ds[i].phat += ds[i].time[j] +
14.                20*(ds[i].nop[j] - 1);
15.                ds[i].sobai ++;
16.            }
17.        }
18.        int kq = 1;
19.        for(int i = 2; i <= n; i++)
20.            if((ds[kq].sobai < ds[i].sobai) || (ds[kq].sobai
21.            == ds[i].sobai && ds[kq].phat > ds[i].phat))
22.                kq = i;
23.        cout<<ds[kq].ten<<" "<<ds[kq].sobai<<" "<<ds[kq].phat;
24.    }

```

## 6. Đổ xe tối ưu

Khi mua sắm trên khu Long Street, Michael thường đổ xe của mình ở một vị trí nào đó và đi bộ vào cửa hàng. Bạn hãy giúp Michael chọn một chỗ đổ xe để khoảng cách phải đi bộ khi mua hàng là nhỏ nhất.

Long Street có thể coi như là một đường thẳng mà tất cả những điểm mua hàng là các điểm có tọa độ nguyên. Bạn sẽ phải trả phí cho mỗi lần đổ xe ở một điểm đổ, điểm đổ là một điểm có tọa độ nguyên. Michael không muốn trả tiền đổ xe nhiều hơn 1 lần và vì anh ta rất khỏe nên có thể mang tất cả các túi xách, hàng hóa mua được giữa các cửa hàng cần đi mà không có vấn đề gì.

### Dữ liệu vào

Dòng đầu tiên chứa một số nguyên  $1 \leq t \leq 100$  là số lượng bộ test. Mỗi bộ test gồm 2 dòng, dòng đầu tiên ghi số cửa hàng  $n$  mà Michael muốn qua mua hàng,  $1 \leq n \leq 20$  và dòng thứ hai ghi  $n$  số nguyên là các điểm này trên phố Long Street,  $0 \leq x_i \leq 99$ .

### Dữ liệu ra

Với mỗi bộ test, in ra trên một dòng khoảng cách nhỏ nhất phải đi bộ với chỗ đỗ xe tối ưu.

INPUT	OUTPUT	
2	152	
4	70	
24 13 89 37		
6		
7 30 41 14 39 42		

```
1. using namespace std;
2. int xe(int n,int x[]){
3.     int max = x[1], min = x[1];
4.     for(int i=2;i<=n;i++){
5.         if(x[i] < min) min = x[i];
6.         if(x[i] > max) max = x[i];
7.     }
8.     return (max-min)*2;
9. }
10. main(){
11.     int test;
12.     cin>>test;
13.     while(test--){
14.         int n,x[100];
15.         cin>>n;
16.         for(int i=1;i<=n;i++)
17.             cin>>x[i];
18.         cout<<xe(n,x)<<"\n";
19.     }
20. }
```

### 7. Trò chơi dò mìn

Trong trò chơi dò mìn, người ta cho trước một ma trận cấp  $n*m$  trong đó có một số quả mìn ở các vị trí nào đó. Nhiệm vụ của người chơi là xác định vị trí của các quả mìn này dựa trên các ô xung quanh.

Trong bài này, bạn hãy viết chương trình chuyển từ ma trận  $n \times m$  mô tả vị trí các quả mìn và các ô trống thành một ma trận khác trong đó với mỗi ô trống sẽ xác định xem có bao nhiêu quả mìn kề với nó.

**Input:** Có nhiều bộ test, mỗi bộ test bắt đầu bằng 2 số nguyên  $n$  và  $m$  ( $1 \leq n, m \leq 100$ ). Tiếp theo đó là ma trận cấp  $n \times m$  trong đó vị trí có mìn đánh dấu bởi ký tự `*` còn không có là dấu chấm (`.`). Dòng cuối cùng của file Input chứa hai giá trị 0.

**Output:** Với mỗi bộ test, in ra màn hình ma trận tương ứng trong đó các vị trí có dấu chấm được thay bằng một số nguyên cho biết số quả mìn kề với ô đó.

Example

**Input:**

3 2

..

.\*

..

5 5

\*.\*.\*

..\*..

\*\*\*\*\*

.....

..\*\*.

0 0

**Output:**

11

1\*

11

\*3\*3\*

36\*63

\*\*\*\*\*

24553

01\*\*1

```

1. using namespace std;
2. main() {
3.     int n[101], m[101], dem = 0;
4.     char s[101][101][101];
5.     while(1) {
6.         cin>>n[dem]>>m[dem];
7.         if(n[dem] == 0 && m[dem] == 0) break;
8.         for(int i = 0; i < n[dem]; i++) {
9.             int j = 0;
10.            for(; j < m[dem]; j++)
11.                cin>>s[dem][i][j];
12.            s[dem][i][j] = '\0';
13.        }
14.        dem++;
15.    }
16.    for(int k = 0; k < dem; k++)
17.        for(int i = 0; i < n[k]; i++) {
18.            for(int j = 0; j < m[k]; j++) {
19.                if(s[k][i][j] == '.') {
20.                    int d = 0;
21.                    if(s[k][i][j - 1] == '*') d++;
22.                    if(s[k][i][j + 1] == '*') d++;
23.                    if(s[k][i - 1][j] == '*') d++;
24.                    if(s[k][i + 1][j] == '*') d++;
25.                    if(s[k][i - 1][j + 1] == '*') d+
26.                    if(s[k][i + 1][j + 1] == '*') d+
27.                    if(s[k][i - 1][j - 1] == '*') d+
28.                    if(s[k][i + 1][j - 1] == '*') d+
29.                    s[k][i][j] = '0' + d;
30.                }
31.                cout<<s[k][i][j];
32.            }
33.            cout<<"\n";
34.        }
35.    }

```

## 8. Phân tích số nguyên

Rất nhiều số nguyên dương có thể được biểu diễn thành tổng của một dãy các số nguyên liên tiếp.

Ví dụ:

$$6 = 1 + 2 + 3$$

$$9 = 5 + 4 = 2 + 3 + 4$$

Tuy nhiên, số 8 thì lại không thể biểu diễn được như vậy. Bài toán đặt ra là với mỗi số nguyên cho trước, hãy đếm xem có thể có bao nhiêu cách biểu diễn số nguyên đó thành tổng của các số nguyên dương liên tiếp.

**Input:** Dòng đầu tiên ghi số bộ test, không lớn hơn 1000. Mỗi bộ test bao gồm thứ tự bộ test, tiếp theo là một số nguyên dương  $n$  nhỏ hơn  $2^{31}$

**Output:** Với mỗi bộ test, in ra màn hình thứ tự bộ test, tiếp theo là số cách biểu diễn tìm được

Example

**Input:**

```
7
1 6
2 9
3 8
4 1800
5 987654321
6 987654323
7 987654325
```

**Output:**

```
1 1
2 2
3 0
4 8
5 17
6 1
7 23
```

```
1. class Chibiettrau {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int test = sc.nextInt();
6.         while (test-- > 0) {
7.             int number = sc.nextInt();
8.             int n = sc.nextInt();
9.             int count = 0;
10.            for (int i = 2; i <= Math.sqrt(2 * n); i++)
11.                if ((2 * n) % i == 0 && (i + 2 * n / i - 1)
% 2 == 0) count++;

```

```

12.         System.out.println(number + " " + count);
13.     }
14. }
15. }

```

### 9. Rick Flag

Trong hành trình giải cứu Amanda Waller, Slipknot đã bỏ trốn và ngay lập tức hấn bị Rick Flag tiêu diệt bằng cách bấm nút kích hoạt bom ở ngay trên tay. Sau một thời gian chiến đấu, Rick Flag không biết màn hình kích hoạt bom của mình còn hoạt động đúng không. Các bạn hãy kiểm tra giúp Rick Flag nhé.

Biết rằng, màn hình kích hoạt bom của Rick Flag có  $n$  nút đại diện cho  $n$  người, mỗi nút gồm 2 trạng thái 0 là đã ấn nút cho nổ bom, 1 là chưa ấn nút. Nếu màn hình còn hoạt động tốt thì trong  $n$  nút sẽ chỉ có duy nhất một nút ở trạng thái 0.

#### Input

Dòng đầu tiên là số nguyên  $n$  ( $2 \leq n \leq 1000$ )

Dòng tiếp chứa  $n$  số nguyên, mỗi số nguyên có giá trị 0 hoặc 1.

#### Output

In ra "YES" nếu màn hình còn hoạt động đúng và "NO" trong trường hợp còn lại.

#### Example

##### Input:

3

1 0 1

##### Output:

YES

```

1. class Main {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         int n = sc.nextInt();
5.         int num = 0;
6.         for (int i = 0; i < n; i++) {
7.             int x = sc.nextInt();
8.             if (x == 0) num++;
9.             if (num == 2) break;
10.        }
11.        if (num == 1) System.out.println("YES");
12.        else System.out.println("NO");
13.    }
14. }

```

### 10. Bat man

Sau khi giải cứu Gotham khỏi Joker và nhận tội thay Harvey Dent, Batman quyết định nghỉ hưu. Trong lúc rảnh rỗi anh ra ngoài bắt Pokemon và thi đấu với các đối thủ để lên level. Hôm nay

anh đến hội trường để thi đấu ,hội trường thi đấu là một ma trận  $R * S$ , mỗi ô dành cho một huấn luyện viên Pokemon, mỗi huấn luyện viên có thể thi đấu với 8 đấu thủ xung quanh.

Vì mãi mê bắt Pokemon, Batman đến hội trường muộn nhất, anh muốn chọn vị trí nào để anh có thể đấu với nhiều đối thủ nhất (nếu vị trí đầy trống), nếu không có vị trí nào trống anh quyết định đi loang quanh để bắt Pokemon tiếp.

Hãy tính số trận đấu có thể diễn ra giữa tất cả các đối thủ.

### Input

Dòng đầu tiên chứa hai số nguyên  $R, S$  ( $1 \leq R, S \leq 50$ ).

R dòng tiếp theo mỗi dòng chứa S ký tự thể hiện hàng thứ R, ký tự '.' Thể hiện chỗ đó chưa có ai, và 'o' nếu đã có người chọn vị trí đó.

### Output

In ra kết quả bài toán.

### Example

#### Test 1:

##### Input:

```
2 3
..o
o..
```

##### Output:

```
2
```

```
1. lass Chibiettrau {
2.     static int R, S;
3.     static String s[];
4.
5.     static int count(int i, int j) {
6.         int count = 0;
7.         for (int t = -1; t <= 1; t++)
8.             for (int k = -1; k <= 1; k++) {
9.                 int u = i + t, v = j + k;
10.                if (u >= 0 && u < R && v >= 0 && v < S
11.                    && !(t == 0 && k == 0) &&
12.                    s[u].charAt(v) == 'o')
13.                        count++;
14.                return count;
15.            }
16.
17.     public static void main(String[] args) {
18.         Scanner sc = new Scanner(System.in);
19.         R = sc.nextInt();
20.         S = sc.nextInt();
21.         s = new String[R];
```



```

22.         for (int i = 0; i < R; i++)
23.             s[i] = sc.next();
24.         int max = 0, iMax = -1, jMax = -1;
25.         for (int i = 0; i < R; i++)
26.             for (int j = 0; j < S; j++)
27.                 if (s[i].charAt(j) == '.' && max < count(i,
j)) {
28.                     iMax = i;
29.                     jMax = j;
30.                     max = count(i, j);
31.                 }
32.         if (iMax != -1 && jMax != -1)
33.             s[iMax] = s[iMax].substring(0, jMax) + "o" +
s[iMax].substring(jMax + 1);
34.
35.         int battle = 0;
36.         for (int i = 0; i < R; i++)
37.             for (int j = 0; j < S; j++)
38.                 if (s[i].charAt(j) == 'o') {
39.                     battle += count(i, j);
40.                     s[i] = s[i].substring(0, j) + "." +
s[i].substring(j + 1);
41.                 }
42.         System.out.println(battle);
43.     }
44. }

```

## 11. Captain Boomerang

Captain Boomerang là người có khả năng rất đặc biệt. Như biệt danh của mình, anh là master trong việc sử dụng những chiếc boomerang. Ngoài ra, Captain Boomerang còn có sở thích rất kỳ lạ đó là thích chơi với các dãy hoán vị. Trong lúc rảnh rỗi, anh có nghĩ ra bài toán này và đem đi đố mọi người trong team.

Cho 1 dãy gồm  $n$  phần tử:  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 5000$ ). Nhiệm vụ là phải chuyển dãy trên về 1 dãy hoán vị  $n$  phần tử, biết trong 1 bước, chỉ được thay đổi giá trị của 1 phần tử. Hỏi xem cần tối thiểu bao nhiêu bước để hoàn thành nhiệm vụ.

Biết rằng 1 dãy hoán vị  $n$  phần tử là hoán vị bất kỳ của dãy số  $1, 2, 3, \dots, n$

Do không ai trong team có sở thích kỳ lạ này nên cũng chẳng ai muốn làm, các bạn hãy giải đáp bài toán này của Captain Boomerang nhé.

### Input

Dòng đầu tiên nhập 1 nguyên số  $n$  ( $1 \leq n \leq 5000$ ).

Dòng tiếp theo nhập dãy gồm  $n$  số  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 5000$ ).

### Output

In ra kết quả bài toán – số bước ít nhất cần thực hiện.

### Example

**Test 1:**

**Input:**

5  
4 5 4 4 1

**Output:**

2

**Test 2:**

**Input:**

2  
1 1

**Output:**

1

Giải thích test 1: 4 5 4 4 1 => 4 5 2 3 1 => cần 2 bước

```
1. public class Main {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         int n = sc.nextInt();
5.         ArrayList<Integer> a = new ArrayList<>();
6.         for (int i = 1; i <= n; i++) {
7.             int j = sc.nextInt();
8.             if (!a.contains(j) && j <= n)
9.                 a.add(j);
10.        }
11.        System.out.println(n - a.size());
12.    }
13. }
```

## 12. Next round

Trong một cuộc thi lập trình, các lập trình viên, có số điểm lớn hơn hoặc bằng thí sinh có vị trí thứ k trên bảng sẽ được vào vòng thi tiếp theo, miễn sao thí sinh có số điểm dương. Cho số điểm các thí sinh, bạn hãy xác định xem có bao nhiêu người được vào vòng tiếp theo nhé.

### Input

Dòng đầu tiên gồm 2 số nguyên n, k ( $1 \leq k \leq n \leq 50$ )

Dòng thứ 2 chứa n số nguyên  $a[1], a[2], \dots, a[n]$  và n số nguyên này tạo thành một dãy không tăng từ trái qua phải.

### Output

Số thí sinh sẽ vào vòng tiếp theo

### Example

**Test 1:**

**Input:**

8 5  
10 9 8 7 7 7 5 5

**Output:**

6

**Test 2:**

**Input:**

4 2

0 0 0 0

**Output:**

0

```
1. class Defeat {
2.     //Try hard 3-5 accepts a day to defeat Cuy and get a cup of
   coffee
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         int n = sc.nextInt(), k = sc.nextInt(), a[] = new int[n];
7.         for (int i = 0; i < n; i++) a[i] = sc.nextInt();
8.         int j = 0;
9.         while (j < k && a[j] > 0) j++;
10.        if (j == 0) {
11.            System.out.println(0);
12.            return;
13.        }
14.        j--;
15.        int l = j;
16.        while (a[l] > 0 && a[l] == a[j] && l < n - 1) {
17.            //        System.out.println("l=" + l);
18.            l++;
19.        }
20.        if (l < n && a[l] == a[j]) l++;
21.        System.out.println(l);
22.    }
23.
24. }
```

### 13 Xây tháp

Bạn có  $n$  hình lập phương và đang muốn xây dựng một kim tự tháp và đỉnh kim tự tháp là 1 hình lập phương, ở mức thứ 2 có  $1 + 2 = 3$  hình lập phương, tương tự như vậy, cấp thứ  $n$  sẽ có  $1 + 2 + \dots + n$  hình lập phương. Vậy với  $n$  hình lập phương trong tay, bạn sẽ xây dựng được kim tự tháp có chiều cao bao nhiêu.

**Input**

Dòng duy nhất chứa số nguyên  $n$  ( $1 \leq n \leq 10^4$ )

**Output**

Chiều cao lớn nhất có thể của kim tự tháp

### Example

**Input:**

25

**Output:**

4

```
1. class Main {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int n = sc.nextInt(), s = 0, i, a[] = new int[2000];
6.         for (i = 1; a[i - 1] <= 10000; i++) {
7.             s += i;
8.             a[i] = a[i - 1] + s;
9.         }
10.        for (int j = 1; j < i - 1; j++)
11.            if (a[j] <= n && a[j + 1] > n) {
12.                System.out.println(j);
13.                break;
14.            }
15.        }
16.    }
```

### 14 Sưu tầm tem

Groot mới bắt đầu sưu tập tem, chỉ có 26 loại tem được kí hiệu từ a -> z.

Cho một xâu là các kí tự từ a->z mô tả cho các lá tem mà Groot sưu tập được, anh đem khoe với Star Lord về bộ sưu tập của mình và được Star Lord tặng thêm một lá tem nữa. Từ đây Groot nghĩ ra một trò hay để đố Drax, vốn chỉ là 1 người rất ngốc. Bài toán mà Groot đố Drax là đếm xem Groot sẽ có bao nhiêu bộ sưu tập khác nhau nếu như thêm lá tem của Star Lord vào bộ sưu tập hiện tại.

Quy tắc thêm đó là đặt lá tem mới vào 1 vị trí bất kì trong thứ tự ban đầu, kể cả ở 2 đầu, những lá tem cũ không được thay đổi thứ tự. Hai bộ sưu tập giống nhau nếu như xâu mô tả của nó hoàn toàn giống nhau.

### Input

Dòng duy nhất chứa xâu mô tả thứ tự các lá tem ban đầu của Groot, có độ dài không vượt quá 20.

### Output

In ra số bộ sưu tập khác nhau có thể có của Groot.

### Example

#### Input:

a

#### Output:

51

Giải thích: các bộ sưu tập khác nhau là 'aa','ab','ac', ...'az','za', ...'ca','ba'.

```
1. class Defeat {
2.     //Try hard 3-5 accepts a day to defeat Cuy and get a cup of
   coffee
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         String s = sc.next();
7.         int n = s.length();
8.         System.out.println(26*(n+1)-n);
9.     }
10.
11. }
```

### 15 Tín hiệu mật

Úi đang tìm hiểu về các loại tín hiệu mật, đặc biệt nhất là loại tín hiệu có tên là GGWP. Loại mã này rất đặc biệt, nó là 1 dãy tín hiệu 1-0, mà nội dung thông điệp được ẩn sau nó thực chất lại là các đoạn tín hiệu 1-0 liên tiếp có đúng k số 1.

Lúi tự hỏi với 1 đoạn mã GGWP, đã có bao nhiêu thông điệp được truyền đi.

#### Input

Dòng đầu tiên gồm số nguyên k ( $0 \leq k \leq 10^6$ )

Trên dòng tiếp theo, gồm một xâu s có độ dài không quá  $10^6$  là đoạn mã GGWP.

#### Output

Số nguyên duy nhất là số thông điệp đã được gửi đi.

### Example

#### Input:

3  
101100110

#### Output:

7

Giải

thích:

Gồm các thông điệp sau: 1011, 10110, 101100, 011001, 11001, 10011, 100110.

```
1.
2. class Main {
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         int k = Integer.parseInt(sc.nextLine());
7.         String s = sc.nextLine();
8.
9.         ArrayList<Integer> ai = new ArrayList<Integer>();
10.        ai.add(-1);
11.        for (int i = 0; i < s.length(); i++)
12.            if (s.charAt(i) == '1') ai.add(i);
13.        ai.add(s.length());
14.
15.        long result = 0;
16.        if (k == 0) {
17.            for (int i = 1; i < ai.size(); i++) {
18.                long temp = (long) (ai.get(i + k) - ai.get(i
19. + k - 1)) - 1;
20.                temp = temp * (temp + 1) / 2;
21.                // System.out.println(" temp="+temp);
22.                result += temp;
23.            }
24.            System.out.println(result);
25.            return;
26.        }
27.
28.        for (int i = 1; i < ai.size() - k; i++)
29.            result += (long) (ai.get(i + k) - ai.get(i + k -
30. 1))
31.                * (ai.get(i) - ai.get(i - 1));
32.        System.out.println(result);
33.    }
34.}
```

## 16 Bóng đá

Hôm nay Ryze xem chung kết V-League cùng đồng đội. Nhưng vì chỉ vào đi vệ sinh có vài phút, anh ấy đã bỏ lỡ kết quả chung cuộc của trận đấu. Tuy nhiên, anh ấy vẫn xem được mô tả sau trận đấu như sau:

Có  $n$  dòng tương ứng với  $n$  bàn thắng, trên mỗi dòng sẽ ghi tên đội bóng ghi bàn thắng đó. Hãy giúp Ryze xác định xem đội nào sẽ thắng (Input đảm bảo sẽ không có kết quả hòa)

### Input

Dòng đầu tiên nhập số  $n$  ( $1 \leq n \leq 100$ ) là số bàn thắng trong trận đấu

N dòng tiếp theo, mỗi dòng là tên đội ghi bàn thắng đó (tên đội là xâu có ít nhất 1 chữ cái, và nhiều nhất 10 chữ cái, chỉ bao gồm chữ cái latin viết hoa). Đảm bảo sẽ không có kết quả hòa, và chỉ có nhiều nhất 2 team

### Output

In ra tên đội thắng cuộc

### Example

**Test 1:**

**Input:**

1

ABC

**Output:**

ABC

**Test 2:**

**Input:**

4

A

A

KUDLJMXCSE

A

**Output:**

A

**Test 3:**

**Input:**

10

W

W

W

W

W

D

D

W

D

D

W

**Output:**

W

```
1. class Defeat {
2.     //Try hard 3-5 accepts a day to defeat Cuy and get a cup of
   coffee
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         int n = sc.nextInt(), a = 0, b = 0;
7.         String x = "", y = "";
8.         for (int i = 0; i < n; i++) {
```

```

9.         String s = sc.next();
10.        if (x.equals("")) x = s;
11.        else if (!x.equals(s) && y.equals("")) y = s;
12.
13.        if (x.equals(s)) a++;
14.        else b++;
15.    }
16.    if (b > a) x = y;
17.    System.out.println(x);
18. }
19.
20. }

```

### 17. Chữ số n

Viết lần lượt các số tự nhiên sát cạnh nhau, ta sẽ được một dãy số vô hạn "12345678910..."

Tìm chữ số thứ n trong dãy số vô hạn đó.

#### Input

Dòng duy nhất chứa số nguyên n ( $1 \leq n \leq 1000$ )

#### Output

Chữ số thứ n.

#### Example

**Input:**

11

**Output:**

0

```

18.    class Main2 {
19.
20.        public static void main(String[] args) {
21.            Scanner sc = new Scanner(System.in);
22.            // while (true) {
23.                int n = sc.nextInt(), i = 0, d = 0;
24.                // break;
25.                while (d < n) {
26.                    i++;
27.                    int x = d;
28.                    if (i < 10)
29.                        d++;
30.                    else if (i >= 10 && i < 100)

```



```

31.         d += 2;
32.     else
33.         d += 3;
34.     if (d == n) {
35.         // System.out.println("i = " + i);
36.         System.out.println(i % 10);
37.         break;
38.     } else if (d > n) {
39.         int tt = n - x;
40.         System.out.println("i = " + i +
41.             " x = " + x);
42.         System.out.println(i.charAt(tt - 1));
43.         break;
44.     } // }
45.     }
46.
47.     }
48. }
49.

```

## 18. Xúc xắc

Í và Tèo đang chơi xúc xắc, mỗi người chọn ra một con số từ 1 đến 6 và sẽ tung xúc xắc để xem giá trị của con xúc xắc gần với con số của người nào hơn. Giá trị x gọi là gần số a hơn số b nếu  $|x - a| < |x - b|$

Giờ đây bạn biết 2 con số mà Tí và Tèo đã chọn, hãy tính xem có bao nhiêu giá trị của xúc xắc mà gần số của Tí hơn, bao nhiêu giá trị mà khoảng cách tới 2 con số đã chọn như nhau và bao nhiêu giá trị mà nó gần với số của Tèo hơn và in ra kết quả lần lượt theo thứ tự như trên.

### Input

Một dòng duy nhất gồm 2 số nguyên a, b ( $1 \leq a, b \leq 6$ )

### Output

Kết quả bài toán.

### Example

#### Input:

2 5

#### Output:

3 0 3

```

1. class Main2 {
2.

```

```

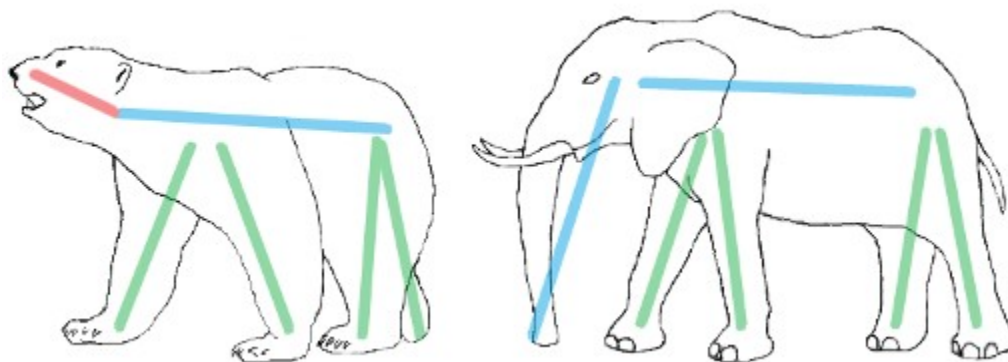
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int a = sc.nextInt(), b = sc.nextInt(), l = 0, t = 0,
        n = 0;
6.         for (int i = 1; i <= 6; i++) {
7.             int x = Math.abs(i - a);
8.             int y = Math.abs(i - b);
9.             if (x > y)
10.                l++;
11.            else if (x == y)
12.                t++;
13.            else
14.                n++;
15.        }
16.        System.out.println(n + " " + t + " " + l);
17.    }
18. }

```

### 19. Xếp que

Hôm nay là 1 ngày đẹp trời, và ngoài thành không có 1 con titan nào. Đội trưởng Levi rảnh rỗi nên đem bó que tính của mình ra chơi, anh có tất cả 6 que tính.

Levi muốn xếp thành hình con gấu hoặc con voi. Biết trong 6 que tính thì 4 que dùng để xếp chân, 1 que dùng để xếp đầu, 1 que dùng để xếp thân. Gấu thì có 4 chân bằng nhau, và đầu phải bé hơn thân. Còn voi thì có 4 chân bằng nhau, và đầu phải dài bằng thân (Minh họa như hình vẽ)



Hỏi xe với 6 que tính được cho, Levi có thể xếp thành hình gấu hoặc voi được không (với điều kiện không được làm gãy bất cứ que tính nào)

Input

Một dòng duy nhất bao gồm 6 số nguyên  $a[1], a[2], \dots, a[6]$  tương đương với độ dài 6 que tính ( $1 \leq a[i] \leq 9$ ). Input đảm bảo rằng không thể ghép cùng lúc được 2 con.

Output



```

21.                                     System.out.println("Elephant");
22.                                     return;
23.                                 }
24.                                     System.out.println("Bear");
25.                                     return;
26.                                 }
27.                                     System.out.println("Alien");
28.                                 }
29.         }
30.

```

## 20. Card game

OOO có n card game, mỗi card được viết trên đó một chữ cái in hoa. Ghost là bạn của OOO, sẽ phải chọn ra k tấm thẻ từ n tấm thẻ. Để thú vị hơn cho việc chọn thẻ thì mỗi cách chọn thẻ sẽ được quy ra một số điểm và Ghost phải chọn k tấm thẻ sao cho số điểm nhận được là lớn nhất. Cách tính điểm như sau.

Điểm = (Số lượng thẻ kí tự 'A')  $^2$  + ... + (Số lượng thẻ kí tự 'Z')  $^2$ .

Các bạn hãy giúp Ghost chọn k thẻ sao cho số điểm nhận được là lớn nhất nhé.

Input

Dòng đầu tiên gồm 2 số nguyên n, k ( $1 \leq k \leq n \leq 10^5$ )

Dòng thứ 2 chứa n kí tự không có dấu cách, kí tự thứ i đại diện cho chữ cái được ghi trên thẻ thứ i ( $1 \leq i \leq n$ )

Output

Số điểm lớn nhất mà Ghost có thể nhận được.

Example

**Input:**

5 3  
AOWBY

**Output:**

3

```

1. class CF {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         long n = sc.nextLong(), k = sc.nextLong();
6.         int a[] = new int[26];
7.         String s = sc.next();
8.         for (int i = 0; i < s.length(); i++) a[s.charAt(i) - 'A']
          ++;
9.         Arrays.sort(a);

```

```

10.         long d = 0;
11.         for (int i = 25; i >= 0 && k > 0; i--) {
12.             long m = Math.min(a[i], k);
13.             d += m * m;
14.             k -= m;
15.         }
16.         System.out.println(d);
17.     }
18. }

```

## 21. Bàn cờ

OOO có một bàn cờ, trên bàn cờ đã được phủ kín bảo 2 loại quân trắng và đen. OOO giờ muốn kiểm tra xem tất cả các ô trên bàn cờ, với các ô kề cạnh của nó có số quân trắng là chẵn.

Bàn cờ được mô tả bởi một mảng 2 chiều kích thước  $n * n$ , quân đen biểu diễn bằng kí tự 'x', quân trắng biểu diễn bằng kí tự 'o'.

In ra "YES" nếu mọi ô trên bàn cờ đều thỏa mãn, in ra "NO" nếu không.

Input

Dòng đầu tiên chứa số nguyên  $n$  ( $1 \leq n \leq 100$ )

$n$  dòng tiếp theo, mỗi dòng chứa  $n$  kí tự biểu diễn cho bàn cờ.

Output

Kết quả bài toán.

Example

**Test 1:**

**Input:**

3

xxo

xox

oxx

**Output:**

YES

**Test 2:**

**Input:**

4

xxxo

xoxo

oxox

xxxx

**Output:**

NO

```

1. class CF {

```

```

2.     static boolean check(int n, String[] s) {
3.         for (int i = 0; i < n; i++)
4.             for (int j = 0; j < n; j++) {
5.                 int d = 0;
6.                 for (int k = -1; k <= 1; k++)
7.                     for (int l = -1; l <= 1; l++) {
8.                         int u = i + k, v = j + l;
9.                         if (!(k == 0 && l == 0) && k * l == 0
10.                            && u >= 0 && u < n && v >= 0
11.                            && s[u].charAt(v) == 'o')
12.                             d++;
13.                     }
14.                 if (d % 2 == 1) return false;
15.             }
16.         return true;
17.     }
18.
19.     public static void main(String[] args) {
20.         Scanner sc = new Scanner(System.in);
21.         int n = sc.nextInt();
22.         String s[] = new String[n];
23.         for (int i = 0; i < n; i++) s[i] = sc.next();
24.         if (check(n, s)) System.out.println("YES");
25.         else System.out.println("NO");
26.     }
27. }
28.

```

## 22. Cắt giấy

Một ngày nọ, Ryze đang ngồi học trong giờ Văn. Do cảm thấy chán và buồn ngủ, nên Ryze đã lôi giấy từ trong cặp sách của mình ra và chơi 1 trò chơi để giết thì giờ.

Ryze có 1 tờ giấy hình chữ nhật kích thước  $a \times b$  ( $a > b$ ). Sau đó anh sẽ gấp chéo 1 tờ giấy, tạo đường nếp có góc 45 độ, 1 mép trùng với 1 cạnh của tờ giấy (như hình vẽ). Sau đó anh ta sẽ cắt phần giấy thừa mà không bị gấp đè lên.



Sau khi cắt, Ryze nhận được 1 mảnh hình vuông  $b \times b$  và 1 mảnh kích thước  $b \times (a-b)$ . Anh tiếp tục làm như vậy với mảnh  $b \times (a-b)$ , và cứ thế cho đến khi tất cả các mảnh đều là hình vuông.

Hãy xác định xem Ryze có thể có được tất cả bao nhiêu mảnh hình vuông.

Input

Một dòng duy nhất bao gồm 2 số nguyên a, b ( $1 \leq b \leq a \leq 10^{12}$ ) – kích thước của miếng giấy chữ nhật ban đầu.

Output

In ra kết quả bài toán – là số hình vuông mà Ryze cắt được.

Example

**Test 1:**

**Input:**

2 1

**Output:**

2

**Test 2:**

**Input:**

10 7

**Output:**

6

**Test 3:**

**Input:**

1000000000000 1

**Output:**

1000000000000

```
1. class Main {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         long a=sc.nextLong(), b=sc.nextLong(),d=0;
5.
6.         while(a>0&&b>0){
7.             long m=Math.min(a, b),M=Math.max(a, b);
8.             d+=M/m;
9.             M=M%m;
10.            a=m;b=M;
11.        }
12.        System.out.println(d);
13.    }
14. }
15.
```

### 23. Kim Cương

Xavi có một ma trận hình vuông kích thước  $n \times n$  (với  $n$  lẻ). Anh ta muốn trang trí vào ma trận đầy một viên kim cương. Các bạn hãy giúp anh ấy thực hiện công việc này.

Kí tự '\*' thể hiện các ô trống còn 'D' thể hiện kim cương (xem ví dụ).

**Input**

Dòng duy nhất chứa số nguyên  $n$  ( $1 \leq n \leq 101$ ).

**Output**

In ra n x n kí tự là đáp án của bài toán.

### Example

**Input:**

5

**Output:**

\*\*D\*\*

\*DDD\*

DDDDD

\*DDD\*

\*\*D\*\*

```
1. */
2. class CF {
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         int n = sc.nextInt();
7.         String s[] = new String[n / 2 + 1];
8.         for (int i = 0; i <= n / 2; i++) {
9.             String t = "", x = "";
10.             for (int j = 0; j < i * 2 + 1; j++) t += "D";
11.             for (int j = 0; j < (n - i * 2 - 1) / 2; j++) x
+= " ";
12.             s[i] = x + t + x;
13.             System.out.println(s[i]);
14.         }
15.         for (int i = n / 2 + 1; i < n; i++)
System.out.println(s[n - i - 1]);
16.     }
17. }
```

### 24. Cặp số đẹp

Một cặp số x, y được gọi là cặp số đẹp cấp m nếu 1 trong 2 số lớn hơn hoặc bằng m. Ví dụ (3, 3) và (0, 2) là cặp đẹp cấp 2, (1, -1) thì không phải vậy.

Bạn có thể biến đổi 1 cặp (x, y) bằng cách xóa 1 trong 2 số và thay vào đó là tổng của 2 số ban đầu.

Hãy đếm xem với tối thiểu bao nhiêu lần biến đổi thì bạn được 1 bộ số đẹp cấp m?

**Input**



Dòng đầu tiên gồm 3 số nguyên x, y, m ( $-10^{18} \leq x, y, m \leq 10^{18}$ ).

### Output

Nếu không có đáp án thỏa mãn in ra -1, ngược lại in ra đáp án.

### Example

Test 1:

Input:

1 2 5

Output:

2

Test 2:

Input:

-1 4 15

Output:

4

Test 3:

Input:

0 -1 5

Output:

-1

```
1. long long goodNum(long long x, long long y, long long m){
2.     long long count = 0;
3.     long long sum = 0;
4.     long long max, min;
5.     if(x > y){
6.         max = x;
7.         min = y;
8.     }else{
9.         max = y;
10.        min = x;
11.    }
12.    if(max >= m){
13.        return 0;
14.    }
15.    if(max <= 0){
16.        return -1;
17.    }
```

```

18.
19.         if(min < 0){
20.             count = -min/max + 1;
21.             min += count * max;
22.         }
23.
24.         while( max < m){
25.             long long temp=max;
26.             max+=min;
27.             min=temp;
28.
29.             count++;
30.         }
31.         return count;
32.     }
33.     main(){
34.         long long x, y, m;
35.         cin>>x>>y>>m;
36.         cout<<goodNum(x, y, m);
37.     }

```

25. Lựa chọn dãy số

Cho dãy  $a[]$  có  $m$  phần tử. Chọn ra  $n$  phần tử sao cho với  $A$  là số lớn nhất trong  $n$  số,  $B$  là số nhỏ nhất trong  $n$  số thì hiệu  $A - B$  là nhỏ nhất trong các cách chọn ra  $n$  số.

Input

Dòng đầu chứa 2 số nguyên  $n, m$  ( $2 \leq n \leq m \leq 50$ )

Dòng thứ 2 chứa  $m$  số nguyên  $a[1], a[2], \dots, a[m]$  ( $1 \leq i \leq m, 4 \leq a[i] \leq 1000$ )

Output

In ra hiệu nhỏ nhất tìm được.

Example

**Input:**

2 10

4 5 6 7 8 9 10 11 12 12

**Output:**

0

## 25. Tìm lại dãy số

Bạn được cho 1 dãy các số nguyên A. Từ dãy số A ban đầu, ta tìm được dãy B theo cách sau:

$$B[i] = (a[1] + a[2] + \dots + a[i]) / i;$$

Nhiệm vụ của bạn là từ dãy B cho trước, hãy tìm lại dãy A

Input

Dòng đầu tiên nhập 1 số tự nhiên n ( $0 < n < 101$ ) là số các phần tử dãy B.

Dòng tiếp theo gồm các số tự nhiên  $B[i]$  ( $1 \leq B[i] \leq 10^9$ ).

Input đảm bảo kết quả luôn là các số nguyên

Output

In ra trên 1 dòng các phần tử của dãy A cách nhau bởi 1 dấu cách.

Example

Test 1:

Input:

1

2

Output:

2

Test 2:

Input:

4

3 2 3 5

Output:

3 1 5 11

Test 3:

Input:

5

1 2 2 3 4

Output:

1 3 2 6 8

```
1. main() {
2.     long long sum = 0, b[208] = {0};
3.     int n;
4.     cin>>n;
5.     for(int i = 1; i <= n; i++){
6.         cin>>b[i];
7.     }
8.     for(int i = 1; i <= n; i++){
9.         cout<<b[i] * i - sum<<" ";
10.        sum += b[i] * i - sum;
11.    }
12. }
```

## 26. Chỉ là sắp xếp

Cho 2 dãy số, dãy  $a[]$  có  $n$  phần tử, dãy  $b[]$  có  $m$  phần tử.

Yêu cầu : In ra  $m$  dòng, dòng thứ  $i$  là số lượng số trong dãy  $a$  nhỏ hơn hoặc bằng  $b[i]$ .

Input

Dòng đầu tiên chứa 2 số  $n, m$  lần lượt là số lượng phần tử của 2 dãy  $a[], b[]$ .

Dòng thứ 2 chứa  $n$  số nguyên  $a[1], a[2], \dots, a[n]$ .

$M$  dòng tiếp theo mỗi dòng chứa 1 số nguyên  $b[1], b[2], \dots, b[m]$

( $1 \leq n, m, a[i], b[i] \leq 10^6$ )

Output

In ra kết quả bài toán.

Example

**Input:**

5 3

1 2 3 4 5

2

3

4

**Output:**

2

3

```

1. void quickSort(int *a, int left, int right){
2.     int pivot = a[(left + right) / 2];
3.     int i = left;
4.     int j = right;
5.     while(i < j){
6.         while(a[i] < pivot){
7.             i++;
8.         }
9.         while(a[j] > pivot){
10.            j --;
11.        }
12.        if(i <= j){
13.            if(i < j){
14.                swap(a[i], a[j]);
15.            }
16.            i++;
17.            j--;
18.        }
19.    }
20.    if(i < right){
21.        quickSort(a, i, right);
22.    }
23.    if(left < j){
24.        quickSort(a, left, j);
25.    }
26. }

27.
28. int binarySearch(int *a, int key, int left, int right){
29.     if(right < left){
30.         return (right + 1);
31.     }
32.     int mid = (left + right) / 2;
33.     if(key >= a[mid] && key < a[mid + 1]){
34.         return (mid + 1);
35.     }else if(key < a[mid]){
36.         return binarySearch(a, key, left, mid - 1);
37.     }else{
38.         return binarySearch(a, key, mid + 1, right);
39.     }
40. }

41.
42. main(){
43.     int n, m, *a, *b;
44.     cin>>n>>m;
45.     a = new int[n + 1];
46.     b = new int[m + 1];
47.     for(int i = 0; i < n; i++){
48.         cin>>a[i];
49.     }

```

```

50.         quickSort(a, 0, n - 1);
51.         for(int i = 0; i < m; i++){
52.             cin>>b[i];
53.         }
54.         for(int i = 0; i < m; i++){
55.             cout<<binarySearch(a, b[i], 0, n - 1)<<endl;
56.         }
57.     }

```

## 27. Chuyển đổi

Cho một số a được viết dưới dạng nhị phân.

Hãy viết chương trình in ra số có giá trị  $17*a$  dưới dạng nhị phân.

Input

Dòng duy nhất là biểu diễn của số a dưới dạng nhị phân ( có độ dài không quá 1000 ).

Output

In ra kết quả dưới dạng nhị phân.

Example

**Input:**

1110010

**Output:**

11110010010

```

1. tring mul17(string number){
2.     int n = number.length();
3.     string first = "0000" + number;
4.     string second = number + "0000";
5.     int borrow = 0;
6.     string result = "";
7.     for(int i = first.length() - 1; i >= 0; i--){
8.         char a = first.at(i);
9.         char b = second.at(i);
10.        char c = (a + b + borrow) - '0';
11.        if(c >= '2'){
12.            borrow = 1;
13.            c = c - 2;
14.        }else{
15.            borrow = 0;
16.        }
17.        result = c + result;

```

```

18.         }
19.         if(borrow == 1){
20.             result = "1" + result;
21.         }
22.         while(result.at(0) == '0' && result.length() > 1){
23.             result.erase(0, 1);
24.         }
25.         return result;
26.     }
27.     main(){
28.         string number = "1";
29.         // while(number != "xxx"){
30.             cin>>number;
31.             cout<<mul17(number);
32.         //     }
33.     }

```

## 28. Ký túc xá

Tourist và Rng mới đỗ PTIT (Posts and Telecommunications Institute of Technology). Để dễ dàng trong việc học, 2 bạn quyết định tìm 1 phòng ký túc xá để ở chung. Trường PTIT có n phòng ký túc xá, mỗi phòng hiện có p người ở, và phòng chỉ chứa được tối đa q người. Hãy giúp họ đếm xem có bao nhiêu phòng mà 2 bạn có thể ở cùng nhau.

### Input

Dòng đầu tiên nhập 1 số tự nhiên n ( $0 < n < 101$ ) là số phòng ký túc xá của trường.

N dòng tiếp theo, mỗi dòng gồm 2 số p, q ( $0 < p \leq q < 101$ ).

### Output

In ra một số nguyên là kết quả của bài toán.

### Example

Test 1:

Input:

```

3
1 1
2 2
3 3

```

Output:

```

0

```

Test 2:

Input:

3

1 10

0 10

10 10

Output:

2

```
1. class CF {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int n = sc.nextInt(), d = 0;
6.         for (int i = 0; i < n; i++) {
7.             int a = sc.nextInt(), b = sc.nextInt();
8.             if (b - a >= 2) d++;
9.         }
10.        System.out.println(d);
11.    }
12. }
13.
```

## 29. Trai hay gái

Phiha đã tới tuổi cập kê. Anh chàng cũng hay bận làm việc nên nhiều khi không đi đâu mà chỉ chat chit trên mạng, Voz là nơi mà cậu hay vào nhất. Một hôm Phiha bắt gặp một user với avatar nữ rất xinh đẹp lại còn biết đánh dota, rất hợp gu với Phiha, rồi đánh liều cậu hẹn gặp user đó ngoài đời thực. Nhưng thật bất ngờ, user có avarta xinh đẹp đó lại là Khuyen bạn cậu. Cậu buồn, và không còn muốn có tình cảm với ai nữa.

Sau việc này, cậu phát hiện ra một đặc điểm để phân biệt giới tính rất hiệu quả và để giúp cho các bạn trai khác không bị như cậu, cậu liền viết một chương trình phân biệt tự động. Chao ôi, cứ mỗi lần định viết, nỗi đau lại quay lại, cậu không chạm nổi vào bàn phím, thôi thì, các bạn viết hộ chương trình hộ Phiha nhé. Biết rằng một user là nam nếu số chữ cái riêng biệt trong usernam là số lẻ, ngược lại sẽ là nữ.

Input

Dòng duy nhất là username (Có độ dài không quá 100, chỉ chứa các chữ cái in thường).

Output

In ra "CHAT WITH HER!" nếu user là nữ, ngược lại in ra "IGNORE HIM!"

Example

**Input:**



wjmbzmr

**Output:**

CHAT WITH HER!

```
1. class CF {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         String s=sc.nextLine();
6.         int a[]=new int[26],t=0;
7.         for (int i = 0; i < s.length(); i++)
8.             a[s.charAt(i)-'a']=1;
9.         for(int i:a)t+=i;
10.        if(t%2==0) System.out.println("CHAT WITH HER!");
11.        else System.out.println("IGNORE HIM!");
12.    }
```

### 30. Sắp xếp

Cho dãy số có  $n$  phần tử gồm  $a[1], a[2], a[3], \dots, a[n]$ . Và một thao tác biến đổi như sau, lấy phần tử cuối cùng của dãy số chuyển lên vị trí đầu tiên, ví dụ:  $a[1], a[2], a[3]$  sẽ thành  $a[3], a[1], a[2]$ .

Yêu cầu: Với thao tác biến đổi như vậy thì dãy  $a[]$  có thể được chuyển thành một dãy không giảm sau một số thao tác hữu hạn nào đó không.

Input

Dòng đầu tiên là số nguyên  $n$  ( $1 \leq n \leq 10^5$ ).

Dòng thứ hai chứa  $n$  số nguyên  $a[1], a[2], a[3], \dots, a[n]$  ( $1 \leq a[i] \leq 10^5$ ).

Output

In ra số thao tác cần thiết để biến đổi dãy số, hoặc in ra -1 nếu không thể biến đổi được dãy số ban đầu thỏa mãn yêu cầu.

Example

**Input:**

2

2 1

**Output:**

1

```

1. class CF {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int n = sc.nextInt(), a[] = new int[n];
6.         for (int i = 0; i < n; i++) a[i] = sc.nextInt();
7.         int end = n - 1, start = 0;
8.         while (end > 0 && a[end] >= a[end - 1]) end--;
9.         while (start < n - 1 && a[start] <= a[start + 1]) start+
10.            +;
11.         //          System.out.println(end + " " + start);
12.         if (end == 0 && start == n - 1)
13.             System.out.println(0);
14.         else if (end - start == 1 && a[0] >= a[n - 1])
15.             System.out.println(n - end);
16.         else System.out.println(-1);
17.     }
18. }

```

### 31. Pangram

Một từ được gọi là pangram nếu tất cả các chữ cái trong bảng chữ cái tiếng Anh đều xuất hiện duy nhất 1 lần.

Bạn hãy kiểm tra xem từ bạn nhập vào có phải pangram hay không.

Input

Dòng thứ 1 số tự nhiên n là độ dài từ nhập vào ( $1 \leq n \leq 100$ ).

1 từ có thể có cả chữ cái viết hoa và viết thường, độ dài n.

Output

In "YES" nếu từ nhập vào là pangram, "NO" trong trường hợp ngược lại.

Example

**Input:**

12

toosmallword

**Output:**

NO

```

1. class CF {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int a[] = new int[26], p = 1;
6.         sc.nextInt();
7.         for (int i = 0; i < 26; i++) a[i] = 0;

```

```

8.         String s = sc.next();
9.         for (int i = 0; i < s.length(); i++) {
10.             char c = s.charAt(i);
11.             if (c >= 'A' && c <= 'Z') a[c - 'A'] = 1;
12.             if (c >= 'a' && c <= 'z') a[c - 'a'] = 1;
13.         }
14.         for (int i : a) p *= i;
15.         if (p == 0) System.out.println("NO");
16.         else System.out.println("YES");
17.     }
18. }

```

### 32. Giờ học lý thú

Trong giờ học về dãy số fibonacci, Magnus đã nghĩ ra 1 dãy số tương tự như sau:

$F(1) = x$ ,  $F(2) = y$ ,  $F(i) = F(i-1) + F(i+1)$  với mọi  $i \geq 2$ .

#### Input

Dòng đầu tiên chứa 2 số nguyên  $x, y$  ( $|x|, |y| \leq 10^9$ ).

Dòng thứ hai chứa số nguyên dương  $n$  ( $n \leq 10^9$ ).

#### Output

Ghi ra giá trị của  $F(n)$ .

#### Example

##### Input:

1 2

3

##### Output:

1

```

1. */
2. class CF {
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         int x = sc.nextInt(), y = sc.nextInt(), n = sc.nextInt(),
           z = 0;
7.         switch (n % 6) {
8.             case 0:
9.                 z = x - y;
10.                break;
11.             case 1:
12.                 z = x;
13.                break;
14.             case 2:

```

```

15.         z = y;
16.         break;
17.     case 3:
18.         z = y - x;
19.         break;
20.     case 4:
21.         z = -x;
22.         break;
23.     case 5:
24.         z = -y;
25.         break;
26.     }
27.     System.out.println(z);
28. }
29. }
30.

```

### 33. Ký tự giống nhau

Cho một chuỗi  $s$  chỉ gồm các ký tự '.' Và '#', có độ dài  $n$  ( $2 \leq n \leq 10^5$ ).

Cho  $m$  truy vấn dạng  $l[i], r[i]$  ( $1 \leq l[i] < r[i] \leq n$ ). Bạn cần tính kết quả của truy vấn là số lượng các giá trị  $k$  ( $l[i] \leq k < r[i]$ ) thỏa mãn  $s[k] = s[k + 1]$ .

#### Input

Dòng đầu tiên là chuỗi  $s$ .

Dòng thứ 2 là số nguyên  $m$  – số truy vấn.

$m$  dòng tiếp theo, dòng thứ  $i$  chứa 2 số nguyên  $l[i]$  và  $r[i]$ .

#### Output

Gồm  $m$  dòng là kết quả của  $m$  truy vấn.

#### Example

Test 1:

Input:

.....

4

3 4

2 3

1 6

2 6

Output:

1  
1  
5  
4

```
1. class Main {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         String s = sc.nextLine();
6.         int n = s.length();
7.         int a[] = new int[n]; a[0]=0;
8.         for (int i = 1; i < n; i++)
9.             if (s.charAt(i-1) == s.charAt(i))
10.                 a[i] = a[i-1]+1;
11.             else
12.                 a[i] = a[i-1];
13.         // for (int i = 0; i < n; i++)
14.         //     System.out.print(a[i] + " ");
15.         int tv = sc.nextInt();
16.         while (tv-- > 0) {
17.             int l = sc.nextInt() - 1, r = sc.nextInt()
18.                 - 1;
19.             System.out.println(a[r] - a[l]);
20.         }
```

### 34. Số nguyên tố kế tiếp

Số nguyên tố y gọi là số nguyên tố kế tiếp của số nguyên tố x nếu y là số nhỏ nhất mà vẫn lớn hơn x.

#### Input

Cho 2 số nguyên x, y ( $2 \leq x < y \leq 50$ ) với x là số nguyên tố.

#### Output

In ra "YES" nếu y là số nguyên tố kế tiếp của x, ngược lại in ra "NO".

#### Example

Test 1:

Input:

3 5

Output:

YES

Test 2:

Input:

7 13

Output:

NO

```
1. class Main {
2.     static boolean a[] = new boolean[51];
3.
4.     static void sang() {
5.         for (int i = 2; i < 51; i++)
6.             a[i] = true;
7.         for (int i = 2; i < 8; i++)
8.             if (a[i])
9.                 for (int j = i + i; j < 51; j += i) {
10.                     a[j] = false;
11.                 }
12.     }
13.
14.     public static void main(String[] args) {
15.         Scanner sc = new Scanner(System.in);
16.         int x = sc.nextInt(), y = sc.nextInt();
17.         boolean t = false;
18.         sang();
19.         if (a[x] && a[y]) {
20.             int i = x + 1;
21.             while (!a[i])
22.                 i++;
23.             if (i == y)
24.                 t = true;
25.         }
26.         if (t)
27.             System.out.println("YES");
28.         else
29.             System.out.println("NO");
30.     }
31. }
```

### 35. XOR

Cho một dãy gồm  $n$  số nguyên không âm. Tìm giá trị lớn nhất có thể khi thực hiện phép tính xor tất cả các phần tử trong một dãy con nào đó của dãy số ban đầu.

**Input**

Dòng đầu tiên là số nguyên  $n$  ( $1 \leq n \leq 100$ ) số phần tử của dãy số.

Dòng thứ 2 chứa n số nguyên là phần tử của dãy số có giá trị nhỏ hơn  $2^{30}$ .

### Output

In ra một số nguyên là kết quả của bài toán.

### Example

Test 1:

Input:

5

1 2 1 1 2

Output:

3

### 36. Đong gạo

Tuyenlv7 bị mẹ giao cho nhiệm vụ đó là đong gạo để mang lên nhà trọ. Anh được mẹ đưa cho 2 loại bịch, là loại 5 kg và 3 kg. Tuyenlv7 sẽ phải đong đủ số gạo mà mẹ cho vào 2 loại bịch trên.

Ví dụ mẹ cho 18 kg thì Tuyenlv7 có thể đong bằng 3 bịch 5kg + 1 bịch 3kg hoặc 6 bịch 3 kg.

Hãy giúp anh ấy đong với số lượng bịch ít nhất có thể, nếu không thể đong được, in ra -1.

### Input

Dòng duy nhất chứa số N là số gạo mẹ Tuyenlv7 cho ( $0 < N < 5000$ ).

### Output

In ra đáp án của bài toán.

### Example

**Input:**

18

**Output:**

4

```
1. class Main {  
2.
```

```

3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.
6.         int n = sc.nextInt(), m = 50000;
7.         boolean b = false;
8.         for (int i = 0; i <= n / 5; i++) {
9.             if ((n - i * 5) % 3 == 0) {
10.                 m = Math.min(m, i + (n - i * 5) / 3);
11.                 b = true;
12.             }
13.         }
14.         if (!b)
15.             System.out.println(-1);
16.         else
17.             System.out.println(m);
18.     }
19. }

```

### 37. Trang trí bàn cờ

Tí đang muốn làm một cái bàn cờ vua mới cho riêng mình. Sau khi được anh trai làm cho một tấm gỗ phẳng và nhẵn, Tí bắt đầu sơn màu cho bàn cờ của mình. Bàn cờ của Tí gồm có  $R \times C$  ô, mỗi ô có chiều cao bằng  $A$  và chiều rộng bằng  $B$ , chúng được tô màu đen, trắng xen kẽ.

Nhiệm vụ của các bạn là hãy mô tả bàn cờ của Tí.

#### Input

Dòng đầu tiên gồm 2 số nguyên  $R$  và  $C$ .

Dòng thứ 2 gồm 2 số nguyên  $A$  và  $B$  ( $A, B, R, C \leq 10$ ).

#### Output

In ra  $R \times A$  hàng, mỗi hàng gồm  $C \times B$  kí tự miêu tả bàn cờ của Tí sau khi được trang trí. Ô màu đen được thể hiện bằng kí tự 'X', màu trắng thể hiện bằng kí tự '.'.

#### Example

Test 1:

Input:

```

2
2 2

```

Output:

```

XX..XX..
XX..XX..
..XX..XX
..XX..XX

```



Test 2:

Input:

5  
2 3

5

Output:

```
XXX...XXX...XXX
XXX...XXX...XXX
...XXX...XXX...
...XXX...XXX...
XXX...XXX...XXX
XXX...XXX...XXX
...XXX...XXX...
...XXX...XXX...
XXX...XXX...XXX
XXX...XXX...XXX
```

```
1. class Main2 {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int r = sc.nextInt(), c = sc.nextInt();
6.         int a = sc.nextInt(), b = sc.nextInt();
7.         char x;
8.         for (int i = 0; i < r; i++) {
9.
10.             for (int j = 0; j < a; j++) {
11.                 if (i % 2 == 0)
12.                     x = 'X';
13.                 else
14.                     x = '.';
15.                 for (int j2 = 0; j2 < c; j2++) {
16.                     for (int k = 0; k < b; k++)
17.                         System.out.print(x);
18.                     if (x == 'X')
19.                         x = '.';
20.                     else
21.                         x = 'X';
22.                 }
23.                 System.out.println();
24.             }
25.         }
26.     }
27.
28. }
29.
```

38. Chiếc vé máy mắ

Một số may mắn là số được tạo ra bởi 2 số 4 và 7. Ví dụ, 47, 44 là số may mắn, còn 13 thì không phải.

Tèo là một người rất thích các chiếc vé xe lửa, mỗi chiếc vé đều có số ghi trên đó và tất cả các số đều có chứa chữ số. Tèo gọi một chiếc vé là chiếc vé may mắn nếu số ghi trên vé đó là số may mắn và tổng các chữ số nửa đầu bằng tổng các chữ số ở nửa cuối.

Các bạn xem hộ Tèo chiếc vé mà cậu có là chiếc vé may mắn không nhé.

Input

Dòng đầu tiên là số nguyên dương  $n$  ( $2 \leq n \leq 50$ ), độ dài của số ghi trên vé của Tèo.

Dòng thứ hai là số ghi trên vé của Tèo.

Output

In ra "YES" nếu đó là chiếc vé may mắn, nếu không in ra "NO".

Example

Test 1:

Input:

2

47

Output:

NO

Test 2:

Input:

4

4738

Output:

NO

Test 3:

Input:

4

4774

Output:

YES

```

1. class Main2 {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         Integer.parseInt(sc.nextLine());
6.         String s = sc.nextLine();
7.         int n = s.length(), A = 0, B = 0;
8.         for (int i = 0; i < n / 2; i++) {
9.             char a = s.charAt(i);
10.            char b = s.charAt(i + n / 2);
11.            if ((a != '4' && a != '7') || (b != '4' &&
12.            b != '7')) {
13.                System.out.println("NO");
14.                return;
15.            }
16.            A += (a - '0');
17.            B += (b - '0');
18.        }
19.        String s1 = "NO";
20.        if (A == B)
21.            s1 = "YES";
22.        System.out.println(s1);
23.    }
24.
25.

```

### 39. Hoán vị kế tiếp

Trong bài này, bạn hãy viết chương trình nhận vào một chuỗi (có thể khá dài) các ký tự số và đưa ra màn hình hoán vị kế tiếp của các ký tự số đó (với ý nghĩa là hoán vị có giá trị lớn hơn tiếp theo nếu ta coi chuỗi đó là một giá trị số nguyên).

Chú ý: Các ký tự số trong dãy có thể trùng nhau.

Ví dụ:

123 -> 132

279134399742 -> 279134423799

Cũng có trường hợp sẽ không thể có hoán vị kế tiếp. Ví dụ như khi đầu vào là chuỗi 987.

#### Dữ liệu vào

Dòng đầu tiên ghi số nguyên  $t$  là số bộ test ( $1 \leq t \leq 1000$ ). Mỗi bộ test có một dòng, đầu tiên là số thứ tự bộ test, một dấu cách, sau đó là chuỗi các ký tự số, tối đa 80 phần tử.

#### Dữ liệu ra

Với mỗi bộ test hãy đưa ra một dòng gồm thứ tự bộ test, một dấu cách, tiếp theo đó là hoán vị kế tiếp hoặc chuỗi "BIGGEST" nếu không có hoán vị kế tiếp.

#### Example

**Input:**

```
3
1 123
2 279134399742
3 987
```

**Output:**

```
1 132
2 279134423799
3 BIGGEST
```

```
1. using namespace std;
2. void xuly(char s[]){
3.     int x[100],n = strlen(s);
4.     for(int j=0;j<n;j++)
5.         x[j] = (int)(s[j]-'0');
6.     int i = n-2;
7.     while(i>=0&& x[i]>=x[i+1]) i--;
8.     if(i<0)
9.         cout<<"BIGGEST\n";
10.    else{
11.        int k = n-1;
12.        while(x[i]>=x[k]) k--;
13.        swap(x[i],x[k]);
14.        int r = i+1,s = n-1;
15.        while(r<s){
16.            swap(x[r],x[s]);
17.            r++;s--;
18.        }
19.        for(int j=0;j<n;j++)
20.            cout<<x[j];
21.            cout<<"\n";
22.    }
23. }
24. main(){
25.     int test;
26.     cin>>test;
27.     while(test--){
28.         int stt;
29.         cin>>stt;
30.         char s[100];
31.         cin>>s;
32.         cout<<stt<<" ";
33.         xuly(s);
34.     }
35. }
```

36.

#### 40. Giá trị của năm

Bạn được yêu cầu viết 1 chương trình tính toán với công việc như sau:  
Cho một năm thuộc đoạn 1900 – 2100, hãy tính giá trị của năm đó.

Ta có giá trị của 1 ngày được tính bởi công thức:

Val = tổng các chữ số của ngày + tổng các chữ số của tháng + tổng các chữ số của năm.

Giá trị của năm bằng tổng giá trị của tất cả các ngày trong năm đó.

Ví dụ giá trị của ngày 4/4/2016 là  $4 + 4 + 2 + 0 + 1 + 6 = 17$ .

#### Input

Dòng duy nhất chứa số năm.

#### Output

In ra duy nhất 1 số là đáp án của bài toán.

#### Example

**Input:**

2016

**Output:**

6891

```
1. class Main {
2.
3.     static boolean nhuan(int n) {
4.         return n % 400 == 0 || (n % 4 == 0 && n % 100 != 0);
5.     }
6.
7.     static int tong(int n) {
8.         int t = 0;
9.         int a[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30,
10.            31};
11.         for (int i = 1; i <= 12; i++)
12.             for (int j = 1; j <= a[i]; j++) {
13.                 String s = j + "" + i + n;
14.                 for (int k = 0; k < s.length(); k++)
15.                     t += (s.charAt(k) - '0');
16.             }
17.         if (nhuan(n)) {
18.             String s = "292" + n;
19.             for (int k = 0; k < s.length(); k++)
20.                 t += (s.charAt(k) - '0');
21.         }
22.     }
23. }
```

```

21.         return t;
22.     }
23.
24.     public static void main(String[] args) {
25.         Scanner sc = new Scanner(System.in);
26.         int n = sc.nextInt();
27.         System.out.println(tong(n));
28.     }
29. }
30.

```

#### 41. Đếm giày

Một ngày Gấu muốn đếm lại xem hiện tại mình đang có bao nhiêu đôi giày. Sau khi kiểm tra, Gấu có n chiếc giày màu đỏ và m chiếc giày màu xanh.

Hiện tại Gấu đang theo một là mỗi ngày, gấu đeo một chiếc giày màu đỏ sang bên chân trái, chân phải thì đeo chiếc giày màu xanh. Gấu ngại giặt giày nên sau mỗi ngày, Gấu không đeo lại giày mà hôm đó đã dùng. Các bạn giúp Gấu xem là Gấu theo một này được bao nhiêu lâu. Sau đó, khi không thực hiện một này được nữa thì Gấu sẽ đeo 2 đôi giày cùng màu thì Gấu sẽ có giày đeo được bao nhiêu ngày tiếp theo.

#### Input

Một dòng duy nhất chứa 2 số nguyên n, m ( $1 \leq n, m \leq 100$ ) là số lượng giày màu đỏ và số lượng giày màu xanh.

#### Output

Gồm 2 số nguyên lần lượt là số ngày Gấu đi mỗi bên một màu và số ngày tiếp theo Gấu đi 2 bên màu giống nhau.

#### Example

Test 1:

Input:

3 1

Output:

1 1

```

1. class Ideone
2. {
3.     public static void main (String[] args) throws
4.         java.lang.Exception
5.     {
6.         Scanner sc=new Scanner(System.in);
7.         int a=sc.nextInt(),b=sc.nextInt();
8.         int g=Math.min(a, b), n = Math.max(a,b)-g;
9.     }
10. }

```

```

8.         System.out.println(g+" "+n/2);
9.
10.        }
11.    }

```

#### 42. Ghép hình

Cho 3 hình chữ nhật có kích thước tương ứng là:  $a[i] \times b[i]$  ( $i=1,2,3$ ). Trong đó  $a[i], b[i]$  là các số nguyên dương không quá  $10^9$ . Hỏi 3 hình chữ nhật trên có thể ghép thành một hình vuông không? Nếu ghép được cho biết độ dài cạnh hình vuông đó, ngược lại ghi số 0.

Input

Gồm 3 dòng, mỗi dòng ghi 2 số nguyên dương là kích thước của một hình chữ nhật

Output

Một dòng duy nhất là đáp số của bài toán

Example

INPUT	OUTPUT
1 6 6 2 3 6	6

INPUT	OUTPUT
1 6 6 2 2 4	0

```

1. */
2. class BaiL {
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         long a[] = new long[4], b[] = new long[4], max[] = new
long[4], min[] = new long[4];
7.         for (int i = 0; i < 3; i++) {
8.             a[i] = sc.nextLong();
9.             b[i] = sc.nextLong();
10.            max[i] = Math.max(a[i], b[i]);
11.            min[i] = Math.min(a[i], b[i]);
12.        }

```

```

13.         if (max[0] == max[1] && max[0] == max[2] && max[0]
== min[1] + min[2] + min[0]) {
14.             System.out.println(max[0]);
15.             return;
16.         } else if (max[0] == max[1] && max[2] == max[0] +
min[2] && max[2] == min[0] + min[1]) {
17.             System.out.println(max[2]);
18.         } else if (max[1] == max[2] && max[0] == max[1] +
min[0] && max[0] == min[2] + min[1]) {
19.             System.out.println(max[0]);
20.         } else if (max[0] == max[2] && max[1] == max[0] +
min[1] && max[1] == min[0] + min[2]) {
21.             System.out.println(max[1]);
22.         } else {
23.             System.out.println(0);
24.         }
25.     }
26. }

```

43. Lập lại các kí tự

### PTIT121I - Lập lại các kí tự

Với một xâu ký tự S, và một số nguyên R, hãy tạo ra một xâu T bằng cách mỗi ký tự trong S được lặp lại R lần. Các ký tự trong S có thể nằm trong dãy:

0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ\$%\*+-.:/:

#### Input

Dòng 1 ghi số bộ test (không quá 1000)

Mỗi bộ test gồm số thứ tự, một dấu cách, tiếp theo là số  $1 \leq R \leq 8$ , một dấu cách, rồi đến xâu ký tự S (không quá 20 ký tự).

#### Output

Với mỗi bộ test ghi trên một dòng số thứ tự bộ test, một dấu cách rồi đến xâu T.

#### Example

##### Input:

```

2
1 3 ABC
2 5 /HTP

```

##### Output:

```

1 AAABBBCCC
2 /////HHHHHTTTTTPPPPP

```



```

1. class BaiI {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         int n = sc.nextInt();
5.         for (int i = 0; i < n; i++) {
6.             int j = sc.nextInt();
7.             int m = sc.nextInt();
8.             String s = sc.next();
9.             System.out.print(j + " ");
10.            for (int k = 0; k < s.length(); k++) {
11.                for (int l = 0; l < m; l++) {
12.                    System.out.print(s.charAt(k));
13.                }
14.            }
15.            System.out.println();
16.        }
17.    }
18.
19. }
20.

```

#### 44. Giai thừa 2

ìm số lần xuất hiện của chữ số x trong số n! (n giai thừa).

Giới hạn:

- $1 \leq n \leq 365$
- x là các chữ số từ 0 đến 9

#### Input

- Dòng đầu là số bộ test T ( $\leq 20$ )
- Sau đó là T dòng, mỗi dòng gồm 2 số nguyên cách nhau bởi dấu cách lần lượt là n và x.

#### Output

- Mỗi bộ test in trên một dòng đáp án.

#### Example

##### Input:

```

2
5 2
7 0

```

##### Output:

```

1
2

```

```

1. class BI {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         int t = sc.nextInt();
5.         while(t-- > 0) {
6.             int n = sc.nextInt();
7.             int x = sc.nextInt();
8.             BigInteger bi = new BigInteger(n + "");
9.             BigInteger r = new BigInteger("1");
10.            for (int i = 1; i <= n; i++) {
11.                BigInteger ii = new BigInteger("" + i);
12.                r = r.multiply(ii);
13.            }
14.            String temp = r.toString();
15.            int d = 0;
16.            for (int i = 0; i < temp.length(); i++) {
17.                if(temp.charAt(i) - '0' == x) {
18.                    d++;
19.                }
20.            }
21.            System.out.println(d);
22.        }
23.    }
24. }
25.

```

#### 45. Đếm số lần lặp

Cho trước  $n$  số nguyên không âm  $a_1, a_2, \dots, a_n$ . Mỗi lần lặp, bạn thay đổi dãy này thành một dãy mới theo cách: phần tử thứ  $k$  trong dãy mới bằng trị tuyệt đối của  $a_k - a_{k+1}$ . Phần tử cuối cùng sẽ là  $a_n - a_1$ . Quá trình lặp sẽ dừng lại khi được một dãy bằng nhau.

Ví dụ với  $n=4$  và bắt đầu với dãy 0 2 5 11 ta sẽ có các lần lặp là:

2 3 6 11

1 3 5 9

2 2 4 8

0 2 4 6

2 2 2 6

0 0 4 4

0 4 0 4

4 4 4 4

Như vậy trong ví dụ này ta sẽ có 8 lần lặp. Hãy viết chương trình các định số lần lặp của một dãy ban đầu

#### Input

Gồm nhiều bộ test, mỗi bộ test gồm 2 dòng:

Dòng 1 ghi số  $n$  ( $2 \leq n \leq 20$ )

Dòng 2 ghi  $n$  số của dãy ban đầu

Input kết thúc khi  $n=0$

### Output

Với mỗi bộ test ghi trên một dòng là số lần lặp theo mẫu dưới đây. Nếu dãy không bằng nhau được sau 1000 lần lặp thì ghi ra dòng “not attained”

### Example

#### Input:

```
4
0 2 5 11
5
0 2 5 11 3
4
300 8600 9000 4000
16
12 20 3 7 8 10 44 50 12 200 300 7 8 10 44 50
3
1 1 1
4
0 4 0 4
0
```

#### Output:

```
Case 1: 8 iterations
Case 2: not attained
Case 3: 3 iterations
Case 4: 50 iterations
Case 5: 0 iterations
Case 6: 1 iterations
```

```
1. using namespace std;
2. int bang(int n, int a[]){
3.     for(int i = 1; i < n; i++)
4.         if(a[i] != a[i + 1]) return 0;
5.     return 1;
```

```

6. }
7. int abs(int x){
8.     if(x >= 0) return x;
9.     else return -x;
10. }
11. main(){
12.     int dem = 1;
13.     while(1){
14.         int n, a[100];
15.         cin>>n;
16.         if(n == 0) break;
17.         for(int i = 1; i <= n; i++) cin>>a[i];
18.         int solan = 0;
19.         while(!bang(n, a) && solan < 1000){
20.             int temp = a[1];
21.             for(int i = 1; i < n; i++)
22.                 a[i] = abs(a[i] - a[i + 1]);
23.             a[n] = abs(a[n] - temp);
24.             solan ++;
25.         }
26.         if(solan == 1000) cout<<"Case "<<dem<<": not
attained\n";
27.         else cout<<"Case "<<dem<<": "<<solan<<
iterations\n";
28.         dem++;
29.     }
30. }
31.

```

#### 46. Tráo bài

```

1. class K {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         int n = sc.nextInt();
5.         int x = sc.nextInt();
6.         int T[] = new int[n + 5];
7.         int S[] = new int[n + 5];
8.         for (int i = 1; i <= n + 1; i++) {
9.             T[i] = i - 1;
10.            S[i] = i + 1;
11.        }
12.        S[0] = 1;
13.        for (int i = 0; i < x; i++) {
14.            int u = sc.nextInt();
15.            int v = sc.nextInt();
16.
17.            T[S[u]] = T[u];
18.            S[T[u]] = S[u];
19.            T[u] = T[v];
20.            S[T[v]] = u;

```

```

21.             T[v] = u;
22.             S[u] = v;
23.         }
24.         int t = 0;
25.         for (int i = 1; i <= n; i++) {
26.             t = S[t];
27.             System.out.print(t + " ");
28.         }
29.     }
30. }
31.

```

#### 47. Chuyển đổi hệ cơ số(Bài dễ)

Cho một số ở hệ nhị phân, chuyển số đó sang hệ bát phân

Input

Một số duy nhất ở hệ nhị phân (không quá 100 chữ số, chữ số đầu tiên luôn là 1)

Output

Số sau khi chuyển sang hệ bát phân.

Example

**Input:**

1010

**Output:**

12

**Input:**

11001100

**Output:**

314

```

1. using namespace std;
2. int trave(char s[]){
3.     char so[8][4] = {"000", "001", "010", "011", "100", "101",
   "110", "111"};
4.     for(int i = 0; i < 8; i++)
5.         if(strcmp(s, so[i]) == 0) return i;
6. }
7. main(){
8.     int kq[100], dem = 0;
9.     char s[101];
10.    cin>>s;

```

```

11.         int i = strlen(s) - 1;
12.         for(; i >= 2; i -= 3){
13.             char t[4];
14.             t[3] = '\0'; t[2] = s[i]; t[1] = s[i - 1]; t[0] =
s[i - 2];
15.             kq[dem] = trave(t); dem++;
16.         }
17.         if(i == 1){
18.             char t[4];
19.             t[3] = '\0'; t[2] = s[i]; t[1] = s[i - 1]; t[0] =
'0';
20.             i -= 3; kq[dem] = trave(t); dem++;
21.         }
22.         if(i == 0){
23.             kq[dem] = 1; dem++;
24.         }
25.         while(dem --) cout<<kq[dem];
26.     }
27.

```

#### 48. Chỗ ngồi

Hàng ghế của rạp chiếu phim có N chỗ ngồi. Giữa 2 ghế có một chỗ để nước, trừ các cặp ghế "tình yêu" là không có gì ở giữa (chỉ có chỗ để nước ở 2 phía ngoài ghế), hơn nữa phía ngoài 2 ghế ngoài cùng cũng có chỗ để nước. Giả sử tất cả N người đều muốn để nước cạnh chỗ ngồi của họ, bạn hãy tính xem có thể có tối đa bao nhiêu người có chỗ để nước.

Kí hiệu ghế bình thường là 'S', ghế "tình yêu" là 'L' (2 ghế tình yêu luôn ở cạnh nhau). Kí hiệu '\*' là chỗ để nước.

Nếu dãy ghế là 'SLLLLSSLL' thì có thể mô tả chỗ để nước như sau: \* S \* L L \* L L \* S \* S \* L L \*

Có tối đa 7 người có thể có chỗ để nước.

#### Input

- Dòng 1: chứa số N ( $1 < N \leq 50$ ) - là số ghế
- Dòng 2: chứa chuỗi N kí tự mô tả dãy ghế, 'L' là ghế "tình yêu", 'S' là ghế bình thường.

#### Output

Số người tối đa có chỗ để nước.

#### Example

##### Input:

```

3
SSS

```

##### Output:

```

3

```

**Input:**

4  
SLLS

```

1. main() {
2.             int n, dem = 1, kt = 1;
3.             char s[100];
4.             cin>>n>>s;
5.             for(int i = 0; i < n; i++){
6.                 if(s[i] == 'S') dem++;
7.                 if(s[i] == 'L'){
8.                     if(kt == 0) dem++;
9.                     kt = 1 - kt;
10.                }
11.            }
12.            cout<<dem;
13.        }

```

49. Cột có tích lớn

ho bảng N hàng, M cột, gồm các số nguyên 32-bit có dấu. Các cột đánh số từ 1-> m, bắt đầu từ bên trái. Gọi  $A[i]$  ( $1 \leq i \leq m$ ) là tích tất cả các số của cột i. Tìm cột i mà  $A[i]$  lớn nhất.

**Input**

Gồm nhiều bộ test, mỗi bộ test có dạng:

- Dòng 1: 2 số nguyên m và n ( $1 \leq m \leq 20, 1 \leq n \leq 1000$ )
- N dòng sau, mỗi dòng chứa m số nguyên có dấu.

Input kết thúc bởi dòng chứa 2 số 0.

**Output**

Với mỗi bộ test, in trên 1 dòng chứa cột i mà  $A[i]$  lớn nhất, nếu có nhiều cột thỏa mãn, in ra cột có chỉ số lớn nhất.

**Example****Input:**

```

3 3
20 10 30
15 20 20
30 30 20
3 2
2 -2 2
2 -2 2

```

0 0

Output:

3  
3

```
1. class Main {
2.     static Scanner sc = new Scanner(System.in);
3.     static int n, m;
4.
5.     static void solve() {
6.         double col[] = new double[m];
7.         double max = 0, f;
8.         for (int j = 0; j < m; j++) {
9.             col[j] = sc.nextLong();
10.            if (max < col[j]) max = col[j];
11.            if (max < -col[j]) max = -col[j];
12.        }
13.        f = max;
14.        for (int i = 1; i < n; i++) {
15.            max = 0;
16.            for (int j = 0; j < m; j++) {
17.                double x = sc.nextDouble();
18.                if (f > 0) col[j] *= x / f;
19.                if (max < col[j]) max = col[j];
20.                if (max < -col[j]) max = -col[j];
21.            }
22.            f = max;
23.        }
24.        int colMax = 0;
25.        for (int j = 1; j < m; j++)
26.            if (col[colMax] <= col[j]) colMax = j;
27.        System.out.println(colMax + 1);
28.    }
29.
30.
31.    public static void main(String[] args) {
32.        while (true) {
33.            m = sc.nextInt();
34.            n = sc.nextInt();
35.            if (n == 0 && m == 0) break;
36.            solve();
37.        }
38.    }
39. }
```

50. Bỏ phiếu

Có một số người tham gia trong một cuộc bầu cử, lá phiếu của mỗi người được ghi lại với kí hiệu như sau:

- Ý nghĩa là đồng ý.



- N nghĩa là không đồng ý.
- P nghĩa là có mặt, nhưng không bỏ phiếu.
- A nghĩa là vắng mặt.

Nhiệm vụ của bạn là từ danh sách trạng thái này, xác định kết quả của cuộc bầu cử : Nếu ít nhất một nửa số người vắng mặt, in ra "need quorum". Ngược lại, cuộc bầu cử được tính. Nếu số người đồng ý lớn hơn số người không đồng ý, in ra "yes". Nếu số người không đồng ý lớn hơn số người đồng ý, in ra "no". Nếu số người đồng ý bằng số người không đồng ý, in ra 'tie'.

Input

- Gồm nhiều bộ test, mỗi bộ test là 1 dòng chứa danh sách trạng thái của một số người tham gia cuộc bầu cử. ( $2 \leq \text{Số người} \leq 70$ )
- Bộ test kết thúc bởi dòng chứa dấu '#'

Output

- Mỗi bộ test in trên 1 dòng chứa kết quả của cuộc bầu cử.

Example

**Input:**

YNNAPYYNY  
YAYAYAYA  
PYPPNNYA  
YNNAA  
NYAAA

#

**Output:**

need

yes  
quorum  
tie  
no

need quorum

```
1. using namespace std;
2. main(){
3.     string s;
4.     cin>>s;
5.     while(s != "#"){
6.         int n=s.length();
7.         int yes=0, no=0, p=0, a=0;
8.         for(int i=0;i<n;i++){
9.             if(s[i] == 'Y') yes++;
10.            if(s[i] == 'N') no++;
11.            if(s[i] == 'P') p++;
12.            if(s[i] == 'A') a++;
13.        }
14.        if(a >= (float)n/2) cout<<"need quorum\n";
15.        else{
```

```

16.             if (yes>no) cout<<"yes\n";
17.             if (no>yes) cout<<"no\n";
18.             if (no == yes) cout<<"tie\n";
19.         }
20.         cin>>s;
21.     }
22. }

```

## 51. Tổ chức kì thi

Một cuộc thi gồm có M nữ và N nam đăng kí. Ban tổ chức cần xếp đội cho các thí sinh theo quy tắc như sau: Mỗi đội gồm 2 nữ và 1 nam. Tuy nhiên, ban tổ chức cần K người để tham gia vào công tác tổ chức (chọn từ số người đăng kí), K người này sẽ không thể chọn đội để thi. Ví dụ: Nếu M=6, N=3, K=2. Ban tổ chức có thể lấy 1 nam và 1 nữ để tham gia tổ chức, còn lại 5 nữ và 2 nam. Ban tổ chức có thể tạo ra 2 đội.

Bạn hãy giúp ban tổ chức tính xem có thể chọn tối đa bao nhiêu đội.

Input

Một dòng chứa 3 số nguyên M,N,K. ( $0 \leq M, N \leq 100$ ), ( $0 \leq K \leq M+N$ )

Output

Số nguyên duy nhất là số đội có thể lập được.

Example

**Input:**

6 3 2

**Output:**

2

**Input:**

2 1 1

**Output:**

0

**Input:**

6 10 3

**Output:**

3

```

1. class Defeat {
2.     //Try hard 3-5 accepts a day to defeat Cuy and get a cup of
   coffee
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         int M = sc.nextInt(), N = sc.nextInt(), K = sc.nextInt();
7.         System.out.println(Math.min(N, Math.min(M / 2, (N + M -
   K) / 3)));
8.     }
9.
10.    }

```

## 52. Bán vé số

Fred thích chơi xổ số và cậu ta mua rất nhiều vé số. Mỗi vé gồm 6 số nguyên duy nhất trong khoảng từ 1 đến 49. Fred muốn có một tập vé số chứa tất cả các số từ 1 đến 49.

Cho trước một tập vé số mà Fred đã mua. Hãy kiểm tra xem tập vé đó có chứa tất cả các số từ 1 đến 49 hay không.

### Input

Gồm nhiều bộ test, mỗi bộ test bắt đầu với một dòng chứa số nguyên  $N$  ( $1 \leq N \leq 100$ ) là số vé mà Fred đã mua.

$N$  dòng tiếp theo mô tả các vé số của Fred với 6 số nguyên.

Dữ liệu vào kết thúc với một dòng chứa số 0.

### Output

ghi trên một dòng chữ "Yes" hoặc "No" cho biết bộ vé của Fred có bao phủ hết các số từ 1 đến 49 hay không.

### Example

#### Input:

```

1
1 2 3 4 5 6
9
1 2 3 4 5 6
10 9 8 7 12 11
13 14 15 16 17 18
19 20 21 22 23 24
25 26 27 28 29 30

```

31 32 33 34 35 36

37 38 39 40 41 42

43 44 45 46 47 48

49 19 34 27 25 13

0

**Output:**

No

Yes

```
1. main() {
2.     while(1) {
3.         int a[100], n, dd = 0;
4.         for(int i=1; i<=49; i++) a[i]=0;
5.         cin>>n;
6.         if(n == 0) break;
7.         for(int i=1; i<=n; i++)
8.             for(int j=1; j<=6; j++) {
9.                 int c;
10.
11.                     cin>>c;
12.                     a[c] = 1;
13.             }
14.         for(int i=1; i<=49; i++)
15.             if(a[i] == 0) {
16.                 cout<<"No\n";
17.                 dd++;
18.                 break;
19.             }
20.         if(dd == 0) cout<<"Yes\n";
21.     }
22. }
```

### 53. Biến đổi chuỗi

Cho chuỗi gồm N chữ cái, các chữ cái là 'A' hoặc 'B'. Có hai loại biến đổi:

- Đổi 1 chữ cái bất kì (A -> B hoặc B->A).
- Đổi K chữ cái đầu tiên của chuỗi (A->B và B->A).

Hãy tìm số phép biến đổi ít nhất để biến chuỗi đã cho thành chuỗi chỉ gồm các chữ cái 'A'.

Input

- Dòng 1: số N ( $1 \leq N \leq 1\,000\,000$ )
- Dòng 2: chuỗi gồm N chữ cái, các chữ cái là 'A' hoặc 'B'

Output

Số phép biến đổi tối thiểu.

Example

**Input :**

4

ABBA

**Output :**

2

**Input :**

5

BBABB

**Output :**

2

**Input :**

12

AAABBBAAABBB

**Output :**

4

54. Find the cow:

Cô bò Bessie đã trốn thoát và đang trốn ở một đồi núi với những đồng cỏ cao. Nông dân John (FJ), người đang muốn tìm kiếm Bessie, đã quyết định bò trên đồng cỏ bằng tay và đầu gối để tìm ra dấu vết của Bessie. Không may thay, ông ta có vấn đề với việc tìm kiếm Bessie từ vị trí thuận lợi này. Dãy cỏ ở trước mặt FJ trông như một chuỗi ngoặc đơn có độ dài  $N$  ( $1 \leq N \leq 50,000$ ); ví dụ:

)(((())())

FJ biết rằng Bessie chân sau của Bessie giống như một cặp dấu mở ngoặc đơn ( $($ , và chân trước của cô ta giống như một cặp dấu đóng ngoặc đơn  $)$ . Vị trí của Bessie có thể được diễn tả bởi một cặp  $x < y$ , trong đó  $($  được tìm ở vị trí  $x$ , và  $)$  được tìm ở vị trí  $y$ . Hãy đếm có bao nhiêu vị trí mà Bessie có thể đang đứng.

**Input**

Dòng 1: Một chuỗi ngoặc đơn có độ dài là  $N$  ( $1 \leq N \leq 50,000$ ).

**Output**

Dòng 1: Số vị trí mà Bessie có thể đứng – Có nghĩa là số cặp  $(x, y)$  khác nhau với  $x < y$  sao cho  $($  được tìm thấy ở  $x$  và  $)$  được tìm thấy ở  $y$ .

**Example**

**Input :**

```
)(((())())())
```

**Output:**

4

Có 4 vị trí có thể của Bessie được thể hiện ở dưới

1. )(((())())())

^^ ^^

2. )(((())())())

^^ ^^

3. )(((())())())

^^ ^^

4. )(((())())())

^^ ^^

Có 4 vị trí có thể của Bessie được thể hiện ở dưới

1. )((())())

^^ ^^

2. )((())())

^^ ^^

3. )((())())

^^ ^^

4. )((())())

^^ ^^

```
1.
2.   class Chibiettrau {
3.       public static void main(String[] args) {
4.           Scanner sc = new Scanner(System.in);
5.           String s = sc.next();
6.           int n = s.length();
7.           boolean left[] = new boolean[n];
8.           int right[] = new int[n + 1];
```

```

9.         int result = 0;
10.
11.         for (int i = 1; i < n; i++)
12.             if (s.charAt(i) == s.charAt(i - 1) &&
s.charAt(i) == '(')
13.                 left[i] = true;
14.                 right[n - 1] = 0;
15.                 for (int i = n - 1; i > 0; i--)
16.                     if (s.charAt(i) == s.charAt(i - 1) &&
s.charAt(i) == ')')
17.                         right[i] = right[i + 1] + 1;
18.                         else right[i] = right[i + 1];
19.
20.                 for (int i = 1; i < n; i++)
21.                     if (left[i]) result += right[i];
22.
23.                 System.out.println(result);
24.             }
25.     }
26.

```

## 55. Xóa bit

Cho số a viết ở hệ cơ số 2. Nhiệm vụ của bạn là phải xóa một chữ số để số còn lại xóa khi xóa là lớn nhất !!!

### Input

Một dòng chứa số a viết ở hệ nhị phân.

### Output

Số lớn nhất sau khi xóa 1 chữ số từ a.

### Example

#### Input:

101

#### Output:

11

```

1. using namespace std;
2. main() {
3.     string s;
4.     getline(cin, s);
5.     int i = 0, n = s.length(), k = 0;
6.     while(i < n) {
7.         if(s[i] == '0') {
8.             k++; break;
9.         }

```

```

10.             i++;
11.         }
12.         if(k == 1) s.erase(i,1);
13.         else s.erase(0,1);
14.         cout<<s;
15.     }

```

## 56. Tổng bình phương

Đề

Cho dãy a gồm có N phần tử là hoán vị các số từ 1 đến N. Tính tổng bình phương các số trong dãy A.

### Input

Gồm một số N duy nhất ( $1 \leq N \leq 10^9$ )

### Output

Tổng bình phương các số trong dãy a.

### Example

**Input:**

2

**Output:**

5

```

1. class Main2 {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         String s = sc.nextLine();
5.         BigInteger a = new BigInteger(s);
6.         BigInteger b = new BigInteger(s);
7.         b = b.add(new BigInteger("1"));
8.         BigInteger c = new BigInteger(s);
9.         c = c.multiply(new BigInteger("2"));
10.        c = c.add(new BigInteger("1"));
11.
12.        a = a.multiply(b).multiply(c).divide(new
    BigInteger("6"));
13.        System.out.println(a);
14.    }
15. }
16.

```



57. Đỗ xe  
Đề:

Luka là ông chủ của 3 chiếc xe tải. Hàng đêm, anh thuê lái xe đến mua nấm ở xưởng nấm. Luka đưa trước tiền cho 3 lái xe của mình. Giá tiền đậu xe tại xưởng nấm như sau:

Nếu có 1 xe đậu thì lái xe trả A đồng/phút.

Nếu có 2 xe đậu thì mỗi lái xe trả B đồng/phút.

Nếu có 3 xe đậu thì mỗi lái xe trả C đồng/phút.

Cho biết thời gian đến và đi của mỗi xe. Hãy xác định xem Luka cần phải đưa bao nhiêu tiền cho các lái xe của mình.

**Input**

3 số a, b, c ( $1 \leq c \leq b \leq a \leq 100$ ). 3 dòng tiếp theo là thời gian đến và đi của từng xe. Các giá trị thời gian đều  $\leq 100$  phút.

**Output**

Số tiền mà Luka cần đưa cho 3 lái xe.

**Example**

Input1:

5 3 1  
1 6  
3 5  
2 8

Output1:

33

```
1. class Main2 {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int a=sc.nextInt();
6.         int b=sc.nextInt();
7.         int c=sc.nextInt();
8.         int result[]=new int[101];
9.         for(int i=0;i<3;i++){
10.             int x=sc.nextInt();
11.             int y=sc.nextInt();
12.             for(int j=x+1;j<=y;j++)result[j]++;
13.         }
14.         int money=0;
15.         for(int i=1;i<=100;i++)
16.             switch (result[i]) {
```

```

17.         case 1:
18.             money+=a;
19.             break;
20.         case 2:
21.             money+=b+b;
22.             break;
23.         case 3:
24.             money+=c+c+c;
25.             break;
26.     }
27.     System.out.println(money);
28. }
29.
30. }

```

58.       Blackjack

Đề:

Cho một tập N quân bài, mỗi quân chứa một số nguyên dương. Bạn cần phải chọn ra ba quân bài sao cho tổng các số trên 3 quân bài gần với số M nhất và không vượt quá M.

### Input

Dòng 1 chứa 2 số N và M. ( $N \leq 100$ ,  $M \leq 500\,000$ )

Dòng 2 chứa N số nguyên dương, mỗi số không quá 100 000.

### Output

In ra tổng 3 quân gần M nhất và không vượt quá M.

Input luôn đảm bảo tồn tại đáp số.

### Example

**Input:**

5 21

5 6 7 8 9

**Output:**

21

```

1. using namespace std;
2. main() {
3.     int n; long m, a[101];
4.     cin>>n>>m;
5.     long tong = 0;

```

```

6.         for(int i = 1; i <= n; i++) cin>>a[i];
7.         for(int i = 1; i <= n - 2; i++)
8.             for(int j = i + 1; j <= n - 1; j++)
9.                 for(int k = j + 1; k <= n; k++)
10.                    if(a[i] + a[j] + a[k] > tong && a[i] +
a[j] + a[k] <= m)
11.                        tong = a[i] + a[j] + a[k];
12.             cout<<tong;
13.         }

59.         Nhìn và đọc

```

Phép biến đổi "nhìn và đọc" được định nghĩa như sau:

Bắt đầu từ số ở đầu tiên X. Với các số giống nhau thì nhóm lại thành một dãy (giả sử có Y số) thì dãy này được chuyển thành YX.

Ví dụ 122344111 đọc là "một 1, hai 2, một 3, hai 4, ba 1", do đó chuỗi 122344111 chuyển thành 1122132431.

## Input

Dòng đầu chứa số bộ test:

Mỗi dòng sau chứa 1 bộ test chứa chuỗi cần biến đổi (không quá 1000 chữ số)

## Output

Với mỗi bộ test, in ra kết quả phép biến đổi

## Example

**Input:**

3

122344111

11111111

12345

3

122344111

1111111111

12345

**Output:**

1122132431

```

60.     using namespace std;
61.     void doc(string s){
62.         int i = 0, dem;
63.         while( i < s.length() ){
64.             int j = i + 1;
65.             dem = 1;
66.             while( j < s.length() ){
67.                 if(s[i] == s[j]){
68.                     dem++;
69.                     j++;
70.                 }
71.                 else break;
72.             }
73.             cout<<dem<<s[i];
74.             i = j;
75.         }
76.     }
77.     main() {
78.         int n;
79.         cin>>n;
80.         string arr[10001];
81.         for(int i = 0; i <= n; i++) cin>>arr[i];
82.         for(int i = 0; i <= n; i++){
83.             doc(arr[i]);
84.             cout<<"\n";
85.         }
86.     }
87.
88.

```

## 60. Tính lãi suất

An đem một số tiền ban đầu  $N$  gửi vào ngân hàng với lãi suất  $L$  (đơn vị %). Sau mỗi năm, tiền lãi sẽ cộng dồn vào  $N$ . Hỏi sau bao nhiêu năm thì An sẽ đạt được số tiền là  $M$ .

Input

Dòng đầu là số bộ test.

Mỗi bộ test có dạng: 3 số cách nhau bởi dấu cách:  $N$   $L$   $M$ .

Lưu ý: tất cả các số đều trong giới hạn số thực (kiểu double) và  $N < M$ .

Output

Với mỗi bộ test in ra kết quả yêu cầu trên một dòng

Example

**Input:**

2  
200.00 6.5 300  
500 4 1000.00

**Output:**

7  
18

```
1. main() {  
2.     int test;  
3.     cin>>test;  
4.     while(test--){  
5.         double n, l, m;  
6.         cin>>n>>l>>m;  
7.         double ls=l/100;  
8.         int dem = 0;  
9.         while(n<m) {  
10.             dem++;  
11.             n *= (1+ls);  
12.         }  
13.         cout<<dem<<"\n";  
14.     }  
15. }  
16.
```

**61. BIT operator**

Buổi học thứ 2 về lập trình cơ sở, các bạn sinh viên năm nhất sẽ được giới thiệu một số toán tử về bit. Trong đó 2 toán tử cơ bản nhất là:

Toán tử ++ sẽ tăng giá trị của biến X lên 1 đơn vị.

Toán tử -- sẽ giảm giá trị của biến X đi 1 đơn vị.

Cuối buổi, thầy giáo có một bài tập nhỏ như sau: cho biến X khởi tạo ban đầu bằng 0 và một chuỗi các toán tử bit như trên, các bạn hãy tính giá trị của biến X sau khi thực hiện hết chuỗi lệnh đó?

Input

Dòng đầu tiên là số nguyên n ( $n \leq 150$ ) là số lượng câu lệnh trong chương trình.

n dòng tiếp theo, mỗi dòng là 1 câu lệnh với biến X. Có 4 dạng câu lệnh đó là:

++X

X++

--X

X--

Output

In ra giá trị của biến X sau khi thực hiện xong chương trình.

Example

**Input:**

2

X++

--X

2

X++

--

**Output:**

0

```
1. public static void main(String[] args) {
2.     Scanner sc = new Scanner(System.in);
3.     int n=Integer.parseInt(sc.nextLine()), k=0;
4.     while(n-->0){
5.         String s=sc.nextLine();
6.         if(s.equals("X++")||s.equals("++X")) k++;
7.         else if(s.equals("X--")||s.equals("--X")) k--;
8.     }
9.     System.out.println(k);
10. }
11. }
```

## 62. Tuần lễ công dân

Sau khi đi nhập học, Tèo rất phấn khởi và bắt đầu ngay việc học ở trường đại học. Tuần học đầu tiên là tuần lễ công dân. Mục tiêu chính của Tèo cũng như các tân sinh viên khác là kết bạn.

Hội trường gồm có  $R \times C$  vị trí chỗ ngồi ( $R$  hàng và  $C$  cột). Mỗi bạn sinh viên sẽ cố gắng làm quen và bắt tay với tất cả những người bạn xung quanh mình. Như vậy, mỗi bạn sẽ làm quen được tối đa là 8 người bạn mới.

Buổi học đã bắt đầu, nhưng không may, Tèo lại đến muộn (là người muộn nhất). Tèo sẽ chọn một vị trí trống sao cho có thể làm quen với nhiều bạn mới nhất có thể. Nếu không có ghế trống nào, Tèo quyết định bùng học. :D

Hãy tính toán số lượng cái bắt tay sẽ được thực hiện trong buổi học đầu tiên này.

Input

Dòng đầu tiên là 2 số nguyên R và C ( $1 \leq R, C \leq 50$ ).

R dòng tiếp theo, mỗi dòng chứa C kí tự. Kí tự '.' thể hiện chỗ ngồi còn trống, và kí tự 'o' thể hiện một sinh viên.

Output

In ra một số nguyên duy nhất là số lượng cái bắt tay sẽ được thực hiện.

Example

Test 1:

Input:

2

..o

o..

Output:

2

Giải thích test 1: Tèo có thể ngồi ở vị trí (1,2) hoặc (2,1). Khi đó có tất cả 2 cái bắt tay được thực hiện.

Test 2:

Input:

2

oo

oo

Output:

6

Giải thích test 2: Không có ghế trống nào nên Tèo quyết định đi về. Tuy nhiên, tại buổi học đó vẫn có

tổng cộng 6 cái bắt tay được diễn ra.

```

1. class Chibiettrau {
2.     static int R, S;
3.     static String s[];
4.
5.     static int count(int i, int j) {
6.         int count = 0;
7.         for (int t = -1; t <= 1; t++)
8.             for (int k = -1; k <= 1; k++) {
9.                 int u = i + t, v = j + k;
10.                if (u >= 0 && u < R && v >= 0 && v < S
11.                    && !(t == 0 && k == 0) &&
s[u].charAt(v) == 'o')
12.                    count++;
13.            }
14.            return count;
15.        }
16.
17.        public static void main(String[] args) {
18.            Scanner sc = new Scanner(System.in);
19.            R = sc.nextInt();
20.            S = sc.nextInt();
21.            s = new String[R];
22.            for (int i = 0; i < R; i++)
23.                s[i] = sc.next();
24.            int max = 0, iMax = -1, jMax = -1;
25.            for (int i = 0; i < R; i++)
26.                for (int j = 0; j < S; j++)
27.                    if (s[i].charAt(j) == '.' && max < count(i, j))
28.                    {
29.                        iMax = i;
30.                        jMax = j;
31.                        max = count(i, j);
32.                    }
33.            if (iMax != -1 && jMax != -1)
34.                s[iMax] = s[iMax].substring(0, jMax) + "o" +
s[iMax].substring(jMax + 1);
35.            int battle = 0;
36.            for (int i = 0; i < R; i++)
37.                for (int j = 0; j < S; j++)
38.                    if (s[i].charAt(j) == 'o') {
39.                        battle += count(i, j);
40.                        s[i] = s[i].substring(0, j) + "." +
s[i].substring(j + 1);
41.                    }
42.            System.out.println(battle);
43.        }
44.    }

```



Test IQ là một vòng khá quen thuộc trong khi tuyển dụng. Công ty X muốn test qua khả năng tư duy của bạn, nên đã đưa ra một bài tập IQ nhỏ như sau:

Bạn nhận được một tấm bìa 4x4, mỗi ô của tấm bìa được sơn màu trắng hoặc đen. Nhà tuyển dụng muốn bạn sơn lại nhiều nhất 1 ô, sao cho tồn tại một hình vuông 2x2 mà màu của chúng là đồng nhất.

Input

Gồm 4 dòng, mỗi dòng gồm 4 kí tự. Kí tự '.' thể hiện rằng ô đó được sơn màu trắng, kí tự '#' thể hiện rằng ô đó được sơn màu đen.

Output

In ra "YES" nếu có thể thực hiện được yêu cầu, in ra "NO" trong trường hợp ngược lại.

Example

Test 1:

Input:

####

.#..

####

....

####

.#..

####

....

Output:

YES

Test 2:

Input:

####

....

####

....

Output:

NO

```
public static void main(String[] args) {  
64.         Scanner sc = new Scanner(System.in);  
65.         String s[] = new String[4];  
66.         for (int i = 0; i < 4; i++) s[i] = sc.nextLine();  
67.         for (int i = 0; i < 4; i++)  
68.             for (int j = 0; j < 4; j++) {  
69.                 char c = s[i].charAt(j);  
70.                 int d = 1;  
71.                 for (int t = 0; t < 2; t++)  
72.                     for (int k = 0; k < 2; k++)  
73.                         if (!(k == 0 && t == 0)  
74.                             && (t + i) < 4 && (k + j) < 4  
75.                             && s[t + i].charAt(k + j) == c)  
76.                             d++;  
77.                 if (d >= 3) {  
78.                     System.out.println("YES");  
79.                     return;  
80.                 }  
81.             }  
82.         System.out.println("NO");  
83.     }
```

#### 64. Đàn kiến

Ngày Valentine, Tí mua socola đem đi tỏ tình với người yêu, nhưng đáng tiếc đã bị bạn nữ kia từ chối. Đành rằng đã mua thì phải ăn, Tí mang về nhà ăn một mình. Để thử hương vị lạ, Tí ăn cùng với cả dâu tây, Tí để vài miếng dâu tây lên thanh socola rồi sẵn thể mà chén. Đang chuẩn bị thưởng thức thì Tí nhận được điện thoại từ Tèo, Tèo rủ Tí sang uống rượu giải sầu vì lí do thất tình tương tự. Tí chạy sang luôn, quên mất thanh socola còn đang dang dở trên bàn.

Đàn kiến nhà Tí tha hồ được bữa. Chúng càn quét theo từng hàng hoặc từng cột một. Nhưng không hiểu sao bọn chúng bị dị ứng với mùi của dâu tây, chúng sẽ không “tấn công” những hàng hoặc cột nào có miếng dâu tây, còn đâu là chúng chén sạch!

Các bạn hãy tính toán xem số lượng miếng socola lũ kiến sẽ ăn được là bao nhiêu?

Input

Dòng đầu tiên gồm 2 số r và c ( $2 \leq r, c \leq 10$ ) là số hàng và số cột của thanh socola.

r dòng tiếp theo, mỗi dòng gồm c kí tự, kí tự '.' biểu diễn 1 miếng socola bình thường, kí tự 'S' biểu diễn miếng socola có miếng dâu tây.

Output

In ra số lượng miếng socola nhiều nhất mà đàn kiến có thể ăn được.

Example

**Input:**

```
3 4
S...
....
..S.
```

**Output:**

8

Hình vẽ giải thích test:

```
1. class MyClass {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         int r = sc.nextInt(), c = sc.nextInt(), row = 0, col = 0;
5.         String s[] = new String[r];
6.         for (int i = 0; i < r; i++)
7.             s[i] = sc.next();
8.         for (int i = 0; i < r; i++) {
9.             int k = 0;
10.            while (k < c && s[i].charAt(k) != 'S') k++;
11.            if (k == c) row++;
12.        }
13.        for (int i = 0; i < c; i++) {
14.            int k = 0;
15.            while (k < r && s[k].charAt(i) != 'S') k++;
16.            if (k == r) col++;
17.        }
18.        System.out.println(row * c + col * r - row * col);
19.    }
20. }
```

## 65. Chia Kẹo

Đi chợ về, mẹ mua một gói kẹo cho anh em Tèo. Mẹ bảo Tèo rằng đợi các em đi học về thì hãy chia, nhưng Tèo không nghe, cứ đòi ăn trước. Tèo chia đều kẹo cho tất cả anh em, nếu như chia không đều thì các phần nhiều hơn (nhiều hơn các phần còn lại 1 cái) sẽ dành cho những đứa bé hơn. Là anh cả nên Tèo chỉ lấy phần nhỏ nhất.

Khi lũ trẻ đi học về, chúng nhận phần của mình, nhưng nghi ngờ anh Tèo là người lấy phần nhiều nhất, vì vậy chúng đem đi hỏi mẹ xem gói kẹo ban đầu có tất cả bao nhiêu cái?

Các bạn hãy tính toán xem số lượng kẹo có thể có ban đầu nhỏ nhất và lớn nhất là bao nhiêu?

Input

Gồm 2 số nguyên  $n$  ( $2 \leq n \leq 15$ ) và  $k$  ( $n \leq k \leq 100$ ) lần lượt là những đứa trẻ trong gia đình của Tèo và số kẹo còn lại sau khi Tèo đã ăn hết phần của mình.

Output

In ra 2 giá trị số kẹo nhỏ nhất và lớn nhất có thể có ban đầu.

Example

Test 1:

Input:

2

5

Output:

9 10

Giải thích test 1:

Nếu ban đầu có 9 cái kẹo, Tèo sẽ lấy phần ít hơn là 4 cái, còn lại 5 cái. Nếu ban đầu có 10 cái kẹo,

Tèo sẽ lấy 1 nửa là 5 cái, còn lại 5 cái cho đứa em.

Test 2:

Input:

3

5

Output:

7 7

```

1. class Chibiettrau {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int n = sc.nextInt();
6.         int k = sc.nextInt();
7.         int candy = k / (n - 1);
8.         if (k % (n - 1) == 0) {
9.             System.out.println((k + candy - 1) + " " + (k +
candy));
10.        } else {
11.            System.out.println((k + candy) + " " + (k +
candy));
12.        }
13.    }
14. }

```

## 66. Lũy Thừa

Lũy thừa bậc  $n$  của  $a$  bằng tích của  $n$  thừa số bằng nhau, mỗi thừa số có giá trị bằng  $a$ .  
Cho trước 2 số nguyên  $a$  và  $b$ , các bạn hãy viết chương trình tính giá trị lũy thừa  $a^b$ .

Input

Gồm nhiều test, mỗi test ghi trên 1 dòng, gồm 2 số nguyên không âm  $a$  và  $b$  ( $a \leq 10^9$  và  $b \leq 10^{18}$ ).

Input kết thúc bởi 2 số 0.

Output

Với mỗi test, ghi ra trên một dòng kết quả phép tính lũy thừa  $a^b$  được lấy dư theo 1000 000 007.

Example

**Input:**

```

2 3
2 4
3 2
0 0

```

**Output:**

```

8
16
9

```

```

1. using namespace std;

```

```

2. int ok(long a,long long b){
3.   if(a==0&&b==0) return 0;
4.   return 1;
5. }
6. main(){
7.   long a,c=1000000007;long long b;
8.   cin>>a>>b;
9.   while(ok(a,b)){
10.      int bi[100],k=0;
11.      int tam[100];
12.      while(b>0){
13.         k++;
14.         tam[k]=b%2;
15.         b/=2;
16.      }
17.      for(int i=1;i<=k;i++)
18.         bi[i]=tam[k+1-i];
19.      long long p=1;
20.      for(int i=1;i<=k;i++){
21.         p=(p*p)%c;
22.         if(bi[i]==1){
23.            p=(p*a)%c;
24.         }
25.      }
26.      cout<<p<<"\n";
27.      cin>>a>>b;
28.   }
29. }
30.

```

## 67. Tung đồng xu

Tí và Tèo rủ nhau chơi trò chơi tung đồng xu. Tâm được 2 người tin tưởng giao làm trọng tài. Trò chơi rất đơn giản. Đầu tiên Tí tung đồng xu  $x$  lần, Tèo tung đồng xu  $y$  lần. Nếu người chơi tung được mặt ngửa thì sẽ được 1 điểm, còn mặt sấp thì không có điểm nào cả. Người chiến thắng là người có điểm cao nhất sau khi kết thúc trò chơi. Nếu điểm số Tí và Tèo bằng nhau, sẽ có một kết quả hòa.

Tại một số điểm Tâm lại quên mất kết quả của trò chơi. Tâm chỉ nhớ rằng Tí lật được mặt ngửa ít nhất  $a$  lần, Tèo thì ít nhất  $b$  lần và người chiến thắng là Tí.

Tâm muốn sử dụng tất cả các thông tin trên để tìm ra tất cả các kết quả có thể của trò chơi mà không bị mâu thuẫn với thông tin đang có.

Các bạn hãy giúp Tâm nhé!

Input

Một dòng duy nhất chứa các số nguyên  $x, y, a, b$  ( $1 \leq a \leq x \leq 100, 1 \leq b \leq y \leq 100$ ).

Output

Dòng đầu tiên là số kết quả của trò chơi có thể có. Các dòng tiếp theo ghi ra cặp số  $p[i]$ ,  $q[i]$  là kết quả thứ  $i$  có thể có của trò chơi.

Các kết quả được in theo thứ tự "tăng dần".

Ta định nghĩa như sau: cặp  $(p_1, q_1)$  nhỏ hơn cặp  $(p_2, q_2)$  nếu  $p_1 < p_2$ , hoặc  $p_1 = p_2$  và  $q_1 < q_2$ .

Example

Test 1:

Input:

3 2 1 1

Output:

3

2

3

3 2

1

1

Test 2:

Input:

2 4 2 2

Output:

0

```
1. main() {
2.     int x,y,a,b;
3.     cin>>x>>y>>a>>b;
4.     int dem=0,c[10000],d[10000];
5.     for(int i=a;i<=x;i++)
6.         for(int j=b;j<=y;j++)
7.             if(j<i) {
8.                 c[dem]=i;
9.                 d[dem]=j;
10.                dem++;
11.            }
12.     cout<<dem;
13.     for(int i=0;i<dem;i++)
14.         cout<<"\n"<<c[i]<<" "<<d[i];
15. }
```

### 68. Đoạn thẳng lớn nhất

Bạn được cho  $n$  đoạn thẳng trên tia  $Ox$ . Đoạn thứ  $i$  được bắt đầu từ điểm  $L[i]$  và kết thúc tại  $R[i]$ . Nhiệm vụ của bạn là tìm trong tập đoạn đã cho đoạn thẳng lớn nhất, bao trùm tất cả các tập đoạn còn lại.

Hãy in ra chỉ số của đoạn thẳng đó, nếu không tồn tại thì in ra -1.

Đoạn  $[a, b]$  được gọi là bao trùm đoạn  $[c, d]$  nếu  $a \leq c \leq d \leq b$ .

Input

Dòng đầu tiên là số nguyên  $n$  ( $1 \leq n \leq 100$ ).

$n$  dòng tiếp theo, mỗi dòng gồm 2 số nguyên  $L[i], R[i]$  ( $1 \leq L[i] \leq R[i] \leq 10^9$ ) biểu diễn đoạn thứ  $i$ .

Các đoạn được đánh số bắt đầu 1.

Output

In ra một số nguyên duy nhất là đáp án của bài toán. Input đảm bảo không có 2 đoạn thẳng nào trùng nhau.

Example

Test 1:

Input:

3	
1	1
2	2
3 3	

Output:

-1

Test 2:

Input:

6	
1	5
2	3
1	10
7	10
7	7
10 10	



Output:

3

```
1. using namespace std;
2. long a[1000][10];
3. int n;
4. long test()
5. {
6.     int vt1=-1,vt2=0;
7.     long min1=a[1][1],max=a[1][2];
8.     for(int i=2;i<=n;i++)
9.     {
10.         if(a[i][1]<=min1) min1=a[i][1];
11.         if(a[i][2]>=max) max=a[i][2];
12.     }
13.     for(int i=1;i<=n;i++)
14.     {
15.         if(a[i][1]==min1&& a[i][2]==max) return i;
16.     }
17.     return -1;
18. }
19. main()
20. {
21.     cin>>n;
22.     for(int i=1;i<=n;i++)
23.         for(int j=1;j<=2;j++)
24.             cin>>a[i][j];
25.     cout<<test()<<endl;
26. }
```

## 69. Modulo

Cho 2 số nguyên A và B, A modulo B là phần dư của A khi chia cho B. Ví dụ, 7, 14, 27 và 38 lần lượt là 1, 2, 0 và 2 theo modulo 3.

Cho trước một dãy số có 10 phần tử. Bạn hãy viết chương trình tính số lượng số giá trị khác nhau trong dãy sau khi lấy modulo 42.

Input

Gồm 10 số nguyên không âm  $\leq 1000$ .

Output

Ghi ra một số nguyên duy nhất là số lượng giá trị có trong dãy số sau khi lấy MOD 42.

Example

Test 1:

Input:

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

Output:

10

Test 2:

Input:

42  
84  
252  
420  
840  
126  
42  
84  
420  
126

Output:

1

Test 3:

Input:

39  
40  
41  
42  
43  
44  
82  
83

84

85

Output:

6

```
1. using namespace std;
2. main() {
3.     int cx[42], dem=0;
4.     for(int i=0; i<42; i++)
5.         cx[i]=1;
6.     for(int i=0; i<10; i++) {
7.         int t;
8.         cin>>t;
9.         if(cx[t%42]==1) {
10.             cx[t%42]=0;
11.             dem++;
12.         }
13.     }
14.     cout<<dem;
15. }
16.
```

#### 70. Quán café

Gia đình Tí mới mở một quán cà phê. Qua thăm dò, Tí đã biết trước ngày khai trương sẽ có  $n$  người tới quán trong ngày hôm đó và biết chính xác thời gian người thứ  $i$  sẽ tới vào lúc  $h[i]$  giờ,  $m[i]$  phút.

Quán cà phê mất không quá 1 phút để phục vụ một khách hàng. Nhưng nếu một vị khách tới mà không nhận được sự phục vụ ngay thì sẽ lập tức bỏ đi.

Ngày khai trương nên gia đình Tí muốn phục vụ được tất cả  $n$  người khách. Quán cà phê cần đảm bảo rằng trong mỗi thời điểm số nhân viên phục vụ không ít hơn số lượng khách.

Các bạn hãy giúp Tí biết số nhân viên ít nhất cần có mà vẫn có thể đảm bảo phục vụ trong ngày khai trương.

Input

Dòng đầu tiên ghi số nguyên  $n$  ( $1 \leq n \leq 10^5$ ).

$n$  dòng tiếp theo, mỗi dòng gồm cặp số nguyên  $h[i]$ ,  $m[i]$  tương ứng là giờ và phút mà người thứ  $i$  tới quán ( $0 \leq h[i] \leq 23$ ,  $0 \leq m[i] \leq 59$ ).

Output

In ra số nhân viên ít nhất mà vẫn đảm bảo phục vụ được tất cả  $n$  người.

Example

Test 1:

Input:

4

8

8

8

8 45

Output:

2

Input:

3

0

10

22 22

Output:

1

```
1. using namespace std;
2. main() {
3.     int n,a[100000],b[100000],dem=0,cx[3600];
4.     cin>>n;
5.     for(int i=0;i<3600;i++)
6.         cx[i]=0;
7.     for(int i=0;i<n;i++){
8.         cin>>a[i]>>b[i];
9.         cx[a[i]*60+b[i]]++;
10.    }
11.    for(int i=0;i<3600;i++)
12.        if(dem<cx[i]) dem=cx[i];
13.    cout<<dem;
14. }
```

## 71. Bút màu

Tí nhờ mẹ đi mua bút màu để chuẩn bị cho giờ vẽ tranh trên lớp. Tí dặn mẹ mua 4 bút màu khác nhau, nhưng mẹ Tí lại quên mất, chỉ nhớ là mua 4 cái bút màu cho Tí.

Về đến nhà, Tí bắt đền mẹ vì đã không mua đủ 4 màu cho Tí. Tí đòi mẹ ra hiệu sách mua thêm, để có đủ 4 màu vẽ cho ngày mai.

Các bạn hãy tính xem mẹ Tí cần mua thêm ít nhất bao nhiêu chiếc bút màu?

Input

Một dòng duy nhất gồm 4 số nguyên  $s_1, s_2, s_3, s_4$  ( $1 \leq s_1, s_2, s_3, s_4 \leq 10^9$ ) thể hiện màu của 4 chiếc bút mà mẹ vừa mới mua cho Tí.

Output

In ra số lượng bút màu ít nhất cần mua thêm cho Tí.

Example

Test 1:

Input:

1 7 3 3

Output:

1

Test 2:

Input:

7 7 7 7

Output:

3

```
1. class Main2 {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int a[]=new int[4];
6.         for(int i=0;i<4;i++)a[i]=sc.nextInt();
7.         ArrayList<Integer>ai=new ArrayList<Integer>();
8.         for(int i=0;i<4;i++) if(!ai.contains(a[i]))ai.add(a[i]);
9.         System.out.println(4-ai.size());
10.
11.     }
12.
13. }
```

72. Luyện tập ACM

Sắp đến ACM PTIT 2014, team 3T gồm Tí, Tèo và Tồ tích cực luyện tập hằng say các contest online trên mạng. Một trong những mục tiêu của đội là tiết kiệm thời gian. Cả team quyết định chỉ tập trung vào các bài toán mà có ít nhất 2 thành viên đã có hướng giải quyết.

Các bạn hãy tính xem số lượng bài toán nhiều nhất mà team 3T có thể giải quyết là bao nhiêu?

Input

Dòng đầu tiên gồm số nguyên  $n$  ( $1 \leq n \leq 1000$ ) là số bài toán.

N dòng tiếp theo gồm mỗi dòng có 3 số chỉ chứa 0 và 1. Cột 1 là của Tí, cột 2 của Tèo, và cột 3 của Tồ. Số 1 thể hiện người đó có phương án giải quyết, và số 0 thể hiện không có phương án.

Output

In ra số bài toán mà team 3T có thể giải quyết nhiều nhất có thể.

Example

Test 1:

Input:

```
3
1           1           0
1           1           1
1 0 0
```

Output:

```
2
```

Test 2:

Input:

```
2
1           0           0
0 1 1
```

Output:

```
1
```

```
1. class Main {
2.     public static void main(String[] args) {
```

```

3.         Scanner sc = new Scanner(System.in);
4.         int d=0,n=sc.nextInt();
5.         for(int i=0;i<n;i++){
6.             int a=sc.nextInt()+sc.nextInt()+sc.nextInt();
7.             if(a>=2) d++;
8.         }
9.         System.out.println(d);
10.    }
11. }
12.

```

### 73. Hình hộp chữ nhật

Bạn có một hình hộp chữ nhật có kích thước nguyên và biết được diện tích của 3 mặt có chung 1 đỉnh. Nhiệm vụ của bạn là tính tổng tất cả các cạnh của hình hộp chữ nhật này.

Input

Gồm một dòng duy nhất chứa 3 số nguyên dương không vượt quá  $10^4$  là diện tích 3 mặt của hình hộp chữ nhật.

Output

In ra màn hình duy nhất một số nguyên là kết quả bài toán.

Example

**Input:**

1 1 1

**Output:**

12

```

1. class Main2 {
2.
3.     static int k(int a, int b, int c) {
4.         return (int) Math.sqrt(a * b / c);
5.     }
6.
7.     public static void main(String[] args) {
8.         Scanner sc = new Scanner(System.in);
9.         int a = sc.nextInt(), b = sc.nextInt(), c = sc.nextInt();
10.        System.out.println(4 * (k(a, b, c) + k(c, b, a) +
11.        k(a, c, b)));
12.    }
13. }

```

### 74. Pha nước cam

Tí rất thích uống nước cam. Đó là lý do tại sao trong tủ lạnh của Tí bất cứ đồ uống nào đều có thành phần là cam. Tí có n đồ uống, đồ uống thứ i có tỉ lệ cam chiếm p[i] %.

Một ngày Tí khát nước đã pha n loại đồ uống lại với nhau và muốn biết tỉ lệ cam chiếm bao nhiêu % sau khi trộn, bạn giúp Tí nhé. N loại đồ uống này khi pha có thể tích là như nhau.

Input

Gồm hai dòng:

- \* Dòng đầu tiên ghi số nguyên n ( $1 \leq n \leq 100$ ) là số đồ uống trong tủ lạnh của Tí.
- \* Dòng thứ 2 chứa n số nguyên p[i] ( $0 \leq p[i] \leq 100$ ) là tỉ lệ phần trăm cam chiếm trong đồ uống thứ i.

Output

In màn hình tỉ lệ phần trăm sau khi pha. Kết quả được làm tròn đến 12 số phần thập phân.

Example

**Input:**

3  
50 50 100

**Output:**

66.666666666667

```
1. class Main2 {  
2.  
3.     public static void main(String[] args) {  
4.         Scanner sc = new Scanner(System.in);  
5.         int n = sc.nextInt(), s = 0;  
6.         for (int i = 0; i < n; i++)  
7.             s += sc.nextInt();  
8.         DecimalFormat df = new DecimalFormat("###.#####");  
9.         System.out.println(df.format(s / (double) n));  
}
```

sau dấu phẩy.

Example

**Input:**

3  
1 1 10  
22 1 10  
1 31 10  
3



```
10 10 100
20 20 50
```

```
    }
}
```

```
10.
```

## 75. Điểm cân bằng

Trong không gian hai chiều cho một tập hợp  $n$  điểm, giả sử mỗi điểm được đặc trưng bởi tọa độ  $(x_i, y_i)$  và trọng lượng  $m_i$ . Moment trọng lượng theo trục  $x$  của tập hợp điểm được xác định theo

công thức:  $M_x = \sum \text{all } m_i * (b - y_i)$ .

Còn moment trọng lượng theo trục  $y$  là:  $M_y = \sum \text{all } m_i * (a - x_i)$ .

Điểm cân bằng của tập hợp điểm trên là điểm có tọa độ  $(a, b)$  sao cho cả hai moment trên đều bằng 0.

### Input

Mỗi bộ test bắt đầu với số nguyên  $n$  là số điểm,  $n$  dòng tiếp theo lần lượt ghi tọa độ  $x_i, y_i$  và giá trị  $m_i$  của từng điểm. Các bộ test ngăn cách bởi một dòng trống.

Input kết thúc khi gặp  $n < 0$ .

### Output

Với mỗi bộ test, ghi ra màn hình thứ tự bộ test và giá trị tọa độ  $a, b$  tìm được. Các giá trị tọa độ được làm tròn đến 2 số

```
10 40 30
```

```
-4
```

### Output:

```
Case 1: 8.00 11.00
```

```
Case 2: 12.78 17.78
```

```
1. class Main2 {
2.     static int k=0;
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
```

```

5.         while(true){
6.             int n = sc.nextInt();
7.             if(n<=0)break;
8.             k++;
9.             float x[]=new float[n],y[]=new float[n],m[]=new
float[n];
10.                for(int i=0;i<n;i++){x[i]=sc.nextFloat();
11.                y[i]=sc.nextFloat();
12.                m[i]=sc.nextFloat();}
13.                solve(n,x,y,m);
14.            }
15.        }
16.
17.        private static void solve(int n, float[] x, float[] y,
float[] m) {
18.            float M=0,XM=0,YM=0;
19.            for(int i=0;i<n;i++)
20.                {M+=m[i];XM+=x[i]*m[i];YM+=y[i]*m[i];}
21.            DecimalFormat df = new DecimalFormat("0.00");
22.            System.out.println("Case "+k+": "+df.format(XM/M)
+" "+df.format(YM/M));
23.        }
24.    }
25.

```

## 76. Hai xâu có các kí tự giống nhau

Bài toán đặt ra là cho trước các cặp xâu ký tự, đều chỉ bao gồm các chữ cái viết thường, hay xác định xem mỗi cặp xâu có được tạo ra từ cùng một bộ ký tự giống nhau hay không.

Chú ý: các ký tự lặp cũng phải được xem xét. Ví dụ: "abc" và "aabbccccc" không được xem là tạo nên bởi cùng một bộ ký tự vì xâu thứ 2 các chữ cái được lặp lại nhiều lần hơn.

### Input

Gồm nhiều bộ test, mỗi bộ test gồm hai dòng ghi hai xâu cần kiểm tra, chỉ bao gồm các ký tự viết thường và không quá 1000 ký tự. Input kết thúc khi gặp hai dòng ghi chữ END.

### Output

Với mỗi bộ test, ghi ra màn hình thứ tự bộ test và kết quả bài toán (same hoặc different).

### Example

#### Input:

```

testing
intestg
abc
aabbccccc
abcabcbbc

```

```
aabbbcccc
abc
xyz
END
END
```

#### Output:

```
Case 1: same
Case 2: different
Case 3: same
Case 4: different
```

```
1. class Main2 {
2.     static int k = 0;
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         while (true) {
7.             k++;
8.             String s1 = sc.nextLine();
9.             String s2 = sc.nextLine();
10.            if (s1.equals("END") && s2.equals("END"))
11.                break;
12.            solve(s1, s2);
13.        }
14.    }
15.
16.    private static void solve(String s1, String s2) {
17.        int a1[] = new int[26], a2[] = new int[26];
18.        for (int i = 0; i < s1.length(); i++)
19.            a1[s1.charAt(i) - 'a']++;
20.        for (int i = 0; i < s2.length(); i++)
21.            a2[s2.charAt(i) - 'a']++;
22.        for (int i = 0; i < 26; i++)
23.            if (a1[i] != a2[i]) {
24.                System.out.println("Case " + k + ":
25.                " + "different");
26.                return;
27.            }
28.            System.out.println("Case " + k + ": " + "same");
29.        }
30.    }
31. }
```

#### 77. Dãy số Hailstone

Dãy số Hailstone bắt đầu từ một số tự nhiên N cho trước, các số tiếp theo sẽ được tạo theo quy tắc:

- Nếu số hiện tại là chẵn thì số tiếp theo sẽ bằng số đó chia cho 2.
- Nếu số hiện tại là lẻ thì số tiếp theo sẽ được nhân lên 3 lần rồi cộng 1
- Khi gặp số 1 thì dãy số kết thúc.

Ví dụ, với  $N = 3$  thì dãy số Hailstone tương ứng sẽ là: 3, 10, 5, 16, 8, 4, 2, 1

Bài toán đặt ra là cho trước  $N$ , hãy tính chiều dài của dãy Hailstone tương ứng.

Input

Mỗi bộ test ghi trên một dòng số  $N$  không quá 100. Input kết thúc với  $N = 0$ .

Output

Với mỗi bộ test, ghi ra màn hình duy nhất một số nguyên là kết quả bài toán.

Example

**Input:**

1  
2  
3  
0

**Output:**

1  
2  
8

```
1. using namespace std;
2. main() {
3.     while(1) {
4.         int n;
5.         cin>>n;
6.         if(n == 0) break;
7.         int dd=1;
8.         while(n > 1) {
9.             dd++;
10.            if(n%2 == 0) n /= 2;
11.            else n = n*3+1;
12.        }
13.        cout<<dd<<"\n";
14.    }
15. }
16.
```

78. Xếp hàng

Trong giờ ăn trưa tại Học viện Công nghệ Bưu chính Viễn thông, có  $n$  sinh viên đang xếp hàng để lấy đồ.

Cảm thấy chán vì phải đứng đợi một mình, vì vậy mỗi sinh viên viết ra mã sinh viên của mình đứng ngay trước và ngay sau của mình. Nếu không có ai đứng trước hoặc không có ai đứng sau thì viết ra 0.

Đột nhiên, xe chở nước sôi đi qua, tất cả sinh viên phải tránh. Khi họ trở lại, họ không nhớ vị trí của mình mà chỉ nhớ mã sinh viên của người đứng trước và người đứng sau.

Hãy giúp các sinh viên PTIT tìm lại vị trí của mình!!!!

#### Input

Dòng đầu tiên gồm số tự nhiên  $n$  ( $2 \leq n \leq 2 \cdot 10^5$ ) – số lượng sinh viên.

$n$  dòng tiếp theo, dòng thứ  $i$  gồm cặp số tự nhiên  $a_i, b_i$  ( $0 \leq a_i, b_i \leq 10^6$ ), với  $a_i$  là mã sinh viên của người đứng trước,  $b_i$  là mã sinh viên của người đứng sau 1 sinh viên nào đó. Nếu không có ai đứng trước hoặc không có ai đứng sau nhập 0.

Mã sinh viên của mỗi sinh viên là khác nhau.

#### Output

Trên 1 dòng, in ra  $n$  số  $x_1, x_2, \dots, x_n$ , danh sách của các sinh viên theo thứ tự ban đầu.

#### Example

##### Input:

4

92 31

0 7

31 0

7 141

##### Output:

92 7 31 141

Mình họa test đề bài.

```
1. using namespace std;
2. int n, *a, *b;
3. int *kq,*so;
4. int *sodau, dau;
5. void init(){
6.     cin>>n;
7.     a=new int[n+1];
8.     b=new int[n+1];
9.     for(int i=1;i<=n;i++) cin>>a[i]>>b[i];
```

```

10.         kq=new int[n+1];
11.         so=new int[1000001];
12.         sodau=new int[3];
13.     }
14.     //sap xep nhanh
15. void sort(int dau, int cuoi){
16.     int i=dau, j=cuoi;
17.     int x=a[(dau+cuoi)/2];
18.     while(i<=j){
19.
20.         while(a[i]<x) i++;
21.         while(x<a[j]) j--;
22.         if(i<=j){
23.             swap(a[i],a[j]);
24.             swap(b[i],b[j]);
25.             i++; j--;
26.         }
27.     }
28.     if(dau<j) sort(dau,j);
29.     if(i<cuoi) sort(i,cuoi);
30. }
31. int tknpA(int x){
32.     int dau=1, cuoi=n, giua;
33.     while(dau<=cuoi){
34.         giua=(dau+cuoi)/2;
35.         if(a[giua]==x) return giua;
36.         else if(x>a[giua]) dau=giua+1;
37.         else cuoi=giua-1;
38.     }
39.     return 0;
40. }
41. void sollan(){//tim ra 2 so xuat hien 1 lan
42.     //1 so la o dau, 1 so la o cuoi
43.     for(int i=1;i<=n;i++){
44.         so[a[i]]++;
45.         so[b[i]]++;
46.     }
47.     int dem=0;
48.     for(int i=1;i<=1000000;i++){
49.         if(so[i]==1){
50.             sodau[++dem]=i;
51.             if(dem==2) return;
52.         }
53.     }
54. }
55. void xuli(){
56.     sort(1,n);
57.     //for(int i=1;i<=n;i++)cout<<a[i]<<" ";
58.     sollan();
59.     //cout<<sodau[1]<<" "<<sodau[2]<<" ";
60.     //so xuat hien 1 lan ma nam ben day A la so dau tien

```

```

61.         if (tknpA(sodau[1]) != 0)
62.             dau=sodau[1];
63.         else dau=sodau[2];
64.         //cout<<"Dau: "<<dau;
65.         int chan=2;
66.         int vt=tknpA(0);
67.         while (chan<=n&&vt!=0) {
68.             kq[chan]=b[vt];
69.             vt=tknpA(b[vt]);
70.             chan+=2;
71.         }
72.         int le=3;
73.         kq[1]=dau;
74.         vt=tknpA(dau);
75.         while (le<=n&&vt!=0) {
76.             kq[le]=b[vt];
77.             vt=tknpA(b[vt]);
78.             le+=2;
79.         }
80.         for(int i=1;i<=n;i++) cout<<kq[i]<<" ";
81.     }
82.     main() {
83.         init();
84.         //cout<<"Nhap xong";
85.         sort(1,n);
86.         xuli();
87.     }

```

## 79. Các cặp giai thừa

Axe chơi một trò chơi với Lina.

Họ định nghĩa hàm  $F(x)$  với số  $x$  nguyên dương là tích giai thừa các chữ số của  $x$ .

Ví dụ  $F(135) = 1! * 3! * 5! = 720$ .

Đầu tiên, họ chọn một số  $a$  có  $n$  chữ số và có ít nhất một chữ số lớn hơn 1, có thể có chữ số không ở đầu. Sau đó họ tìm một số nguyên dương  $x$  lớn nhất thỏa mãn:

1.  $x$  không chứa chữ số 0 hoặc 1
2.  $F(x) = F(a)$

Hãy giúp Axe và Lina tìm ra được số đó.

Input

Dòng đầu tiên chứa số bộ test  $T$  ( $T < 100$ ).

Mỗi test gồm một dòng chứa số  $n$  và số  $a$  ( $1 \leq n \leq 15$ ).

Output

In ra kết quả mỗi test trên một dòng là số lớn nhất tìm được.

Example

**Input:**

1

4 1234

**Output:**

33222

```
1. using namespace std;
2. int a[100];
3. void xong(char s[]){
4.     int n = strlen(s), d = 0;
5.     for(int i = 0; i < n; i++){
6.         switch(s[i]){
7.             case '2':
8.                 a[d++] = 2;break;
9.             case '3':
10.                 a[d++] = 3;break;
11.             case '4':
12.                 a[d++] = 2;a[d++] = 2;a[d++] =
13.                 3;break;
14.             case '5':
15.                 a[d++] = 5;break;
16.             case '6':
17.                 a[d++] = 3;a[d++] = 5;break;
18.             case '7':
19.                 a[d++] = 7;break;
20.             case '8':
21.                 a[d++] = 7;a[d++] = 2;a[d++] =
22.                 2;a[d++] = 2;break;
23.             case '9':
24.                 a[d++] = 7;a[d++] = 3;a[d++] =
25.                 3;a[d++] = 2;break;
26.         }
27.     }
28.     for(int i = 0; i < d;i++)
29.         for(int j = 0; j < d;j++)
30.             if(a[i]>a[j]) swap(a[i],a[j]);
31.     cout<<"\n";
32.     for(int i = 0; i<d;i++) cout<<a[i];
33. }
34. int main(){
35.     int n, t;
36.     char s[101][20];
```



```
35.         cin>>n;
36.         for(int i = 0; i < n; i++) cin>>t>>s[i];
37.         for(int i = 0; i<n;i++) xong(s[i]);
38.
39.     }
40.
```

## 80. Số ma thuật

Một số ma thuật là số mà được ghép bởi các số 1, 14, 144. Số ma thuật không nhất thiết phải được ghép bởi cả 3 số trên. Các bạn giúp kiểm tra giúp xem một số có là số ma thuật không nhé!

Input

Một dòng duy nhất chứa số  $n$  ( $1 \leq n \leq 10^9$ ).

Output

In ra "YES" nếu  $n$  là số ma thuật và "NO" trong trường hợp ngược lại.

Example

Test 1:

Input:

114144

Output:

YES

Test 2:

Input:

111111

Output:

YES

Test 3;

Input:

441231

Output:

NO

```
1. using namespace std;
2. bool yesNo(char s[]){
3.     for(int i = 0; i < strlen(s); i++){
4.         if(s[i] != '1' && s[i] != '4') return false;
5.         if(s[i] == '4'){
6.             if(s[i-1] == '1') continue;
7.             else if(s[i-2] == '1') continue;
8.             else return false;
9.         }
10.    }
11.    return true;
12. }
13. main(){
14.     char s[13];
15.     cin>>s;
16.     if(yesNo(s)) cout<<"YES";
17.     else cout<<"NO";
18. }
19.
```

### 81. Chữ số cuối cùng

Thấy Tí rất thích thú với những con số, cô giáo đã giao cho Tí một bài tập về rút gọn các con số. Phép rút gọn được thực hiện như sau: từ một số ban đầu, số mới được tạo thành bằng cách cộng các chữ số của số ban đầu với nhau. Sau đó Tí phải thực hiện tiếp tục với con số vừa mới thu được.

Quá trình rút gọn kết thúc khi số thu được chỉ có duy nhất 1 chữ số.

Các bạn hãy cùng Tí đi tìm chữ số cuối cùng của phép rút gọn!

Input

Dòng đầu tiên gồm số lượng test T ( $T \leq 100$ ).

T dòng tiếp theo, mỗi dòng gồm một số nguyên dương ( $\leq 10^9$ ).

Output

Hãy in ra chữ số cuối cùng sau khi thực hiện phép rút gọn.

Example

**Input:**

3  
1009

167  
102

**Output:**

1  
5  
3

Giải thích:  $167 = 1+6+7 = 14 = 1+4 = 5$

```
1. using namespace std;
2. int cuoi(long long n){
3.     int s = 0;
4.     while(n > 0){
5.         s += n%10;
6.         n /= 10;
7.     }
8.     if(s < 10) return s;
9.     else return (cuoi(s));
10. }
11. main(){
12.     long long a[101];
13.     int n;
14.     cin>>n;
15.     for(int i = 0; i < n; i++) cin>>a[i];
16.     for(int i = 0; i < n; i++) cout<<cuoi(a[i])<<"\n";
17. }
18.
```

## 82. Min Max

Cho số tự nhiên  $m$  và số nguyên  $s$  không âm. Nhiệm vụ của bạn là tìm số bé nhất và lớn nhất có  $m$  chữ số và tổng chữ số bằng  $s$ .

Input

Dòng đầu gồm 2 số  $m$  và  $s$  ( $1 \leq m \leq 100, 0 \leq s \leq 900$ ).

Output

In ra kết quả của bài toán.

Số đầu tiên là số bé nhất, số thứ hai là số lớn nhất. Nếu không có đáp án in ra "-1 -1".

Example

**Input:**

2 15

## Output:

69 96

```
1. using namespace std;
2. int m, s;
3. main(){
4.     cin>>m>>s;
5.     if(s <= 9 && m == 1) cout<<s<<" "<<s;
6.     else if(s == 0 || s > m * 9) cout<<"-1 -1";
7.     else if(s > 9){
8.         int k = s/9, r = s%9;
9.
10.         if(m - k == 1 && r != 0){
11.             cout<<r;
12.             for(int i = 0; i < k; i++) cout<<9;
13.         }
14.         else{
15.             if(r != 0){
16.                 cout<<1;
17.                 for(int i = 0; i < m - k - 2; i++)
18.                     cout<<r - 1;
19.                 for(int i = 0; i < k; i++) cout<<9;
20.             }
21.             else if(r == 0 && m > k){
22.                 cout<<1;
23.                 for(int i = 0; i < m - k - 1; i++)
24.                     cout<<8;
25.                 for(int i = 0; i < k - 1; i++)
26.                     cout<<9;
27.             }else if(r == 0 && m == k){
28.                 for(int i = 0; i < k; i++) cout<<9;
29.             }
30.             cout<<" ";
31.             for(int i = 0; i < k; i++) cout<<9;
32.             if(m - k > 0){
33.                 cout<<r;
34.                 for(int i = 0; i < m - k - 1; i++) cout<<0;
35.             }
36.         }else if(s <= 9){
37.             cout<<s;
38.             for(int i = 0; i < m - 1; i++) cout<<0;
39.             cout<<" 1";
40.             for(int i = 0; i < m - 2; i++) cout<<0;
41.             cout<<s-1;
42.         }
43. }
```

### 83. Hoán vị

Tí rất tích chức năng đệ quy, cậu liền viết một hàm  $f(x)$  để sắp xếp một hoán vị  $n$  số về một dãy số tăng dần. Hàm  $f$  hoạt động như sau.

- Nếu  $x = 1$ , thì thoát ra khỏi hàm  $f$ .
- Nếu không, thì gọi  $f(x - 1)$  và sau đó đổi chỗ  $a[x - 1]$  và  $a[x]$ .

Tuy nhiên hàm  $f$  của Tí hoạt động sai trong một số trường hợp, các bạn đưa ra thử một hoán vị ban đầu mà hàm  $f$  của Tí vẫn đúng nhé.

Input

Một dòng duy nhất chứa số nguyên  $n$  – số lượng phần tử của hoán vị ( $n \leq 1000$ ).

Output

Chứa hoán vị mà hàm  $f$  của Tí vẫn đúng.

Example

**Input:**

2

**Output:**

2 1

```
1. using namespace std;
2. int n, a[1001];
3. void f(int n){
4.     if(n == 1) return;
5.     swap(a[n], a[n-1]);
6.     f(n - 1);
7. }
8. void fa(int n){
9.     if(n == 1) return;
10.    fa(n - 1);
11.    swap(a[n], a[n-1]);
12. }
13. main(){
14.    cin>>n;
15.    for(int i = 1; i <= n; i++) a[i] = i;
16.    cout<<"\n";
17.    f(n);
18.    for(int i = 1; i <= n; i++) cout<<a[i]<<" ";
19. }
```

### 84. Quyết chiến

Có một cuộc quyết chiến giữa 2 phe Radiant và Dire. Mỗi phe có N chiến binh, mỗi chiến binh đều biết chỉ số sức mạnh của mình. Cuộc quyết chiến giữa 2 phe phải được tuân thủ luật sau:

Có N vòng đấu, mỗi vòng đấu Radiant và Dire cử ra 1 chiến binh để quyết chiến với bên kia. Chiến binh của bên nào thắng thì bên đó sẽ được 1 điểm, bên thua cuộc được 0 điểm.

Trước khi cuộc quyết chiến diễn ra, thủ lĩnh của cả 2 phe đều biết được thực lực các chiến binh của đối phương. Thủ lĩnh Dire là 1 người tài trí hơn người, ông ta biết cách sắp xếp thứ tự thi đấu sao cho bên mình có thể đạt được nhiều điểm nhất có thể.

Bạn hãy đoán xem với danh sách 2 đội đã cho trước, Dire có thể có tối đa bao nhiêu điểm.

#### Input

Dòng đầu tiên chứa số N ( $1 \leq n \leq 2 \cdot 10^5$ ).

Dòng thứ 2 gồm n số nguyên dương  $a[1], a[2], \dots, a[n]$  là chỉ số sức mạnh của các chiến binh phe Radiant.

Dòng thứ 3 gồm n số nguyên dương  $b[1], b[2], \dots, b[n]$  là chỉ số sức mạnh của các chiến binh phe Dire.

( $1 \leq a_i, b_i \leq 2 \cdot N$ ).

Giả sử rằng chỉ số sức mạnh của tất cả các chiến binh 2 phe đều khác nhau.

#### Output

In ra số điểm tối đa Dire có thể đạt được.

#### Example

Test 1:

Input:

```
3
3           4           5
1 2 6
```

Output:

1

Test 2:

Input:

```

4
4           5           6           2
1 7 3 8
Output:
3

```

```

1. using namespace std;
2. main() {
3.     int *a,*b, n;
4.     cin>>n;
5.     a=new int[n+1];
6.     b=new int[n+1];
7.     for(int i=0;i<n;i++) cin>>a[i];
8.     for(int i=0;i<n;i++) cin>>b[i];
9.     sort(a,a+n);
10.    sort(b,b+n);
11.    /*for(int i=0;i<n;i++) cout<<a[i]<<" ";
12.    cout<<"\n";
13.    for(int i=0;i<n;i++) cout<<b[i]<<" ";
14.    cout<<"\n";*/
15.    int da=n-1,db=n-1,dem=0;
16.    while(da>=0&&db>=0) {
17.        if(b[db]>a[da]) {
18.            //cout<<a[da]<<" "<<b[db]<<"\n";
19.            dem++;da--;db--;
20.        }else da--;
21.    }
22.    cout<<dem;
23. }

```

## 85. Chẵn lẻ

Tí đang ngồi học toán nhưng thấy chán quá nên ngồi viết ra các số tự nhiên từ 1 tới n và để tăng phần thú vị, đầu tiên, Tí viết các số lẻ trong n số theo thứ tự tăng dần từ trái qua phải ra trước rồi viết tất cả các số chẵn cũng theo thứ tự tăng dần ra sau.

Sau đó Tí đổ tèo biết được giá trị của số thứ k trong dãy số mà Tí vừa viết. Tèo bí quá đành nhờ các bạn giúp vậy.

Input

Một dòng duy nhất chứa 2 số nguyên n và k ( $1 \leq k \leq n \leq 10^{12}$ ).

Output

Giá trị của số thứ k.

Example

Test 1:

Input:

10 3

Output:

5

Test 2:

Input:

7 7

Output:

6

```
1. using namespace std;
2. long long n, k;
3. void xuly() {
4.     if(k <= n/2) {
5.         cout << 1 + (k-1) * 2;
6.         return;
7.     }
8.     if(k == n/2 + 1 && n%2 == 1) {
9.         cout << 1 + (k-1) * 2;
10.        return;
11.    }
12.    long long vt = n/2;
13.    if(n%2 == 1) vt++;
14.    cout << (k - vt) * 2;
15. }
16. main() {
17.     cin >> n >> k;
18.     xuly();
19. }
20.
```

## 86. Mã hóa chuỗi

Tí đang tìm hiểu một trong những kĩ thuật mật mã hóa đơn giản nhất. Với bản tin cần được mã hóa, phương pháp này được sẽ mã hóa thành một chuỗi với quy tắc như sau:

Chuỗi mã hóa chỉ lưu lại các kí tự chữ cái xuất hiện đầu tiên. Nói cách khác, các kí tự nào xuất hiện > 1 lần trở lên, sẽ bị xóa bỏ, chỉ giữ lại kí tự đầu tiên.

Giá trị mật mã của chuỗi được tính bằng tổng số lần xuất hiện của các kí tự có tần suất > 1 lần.

Các bạn hãy cùng Tí giải quyết bài toán này nhé!



## Input

Dòng đầu tiên gồm số lượng test T ( $T \leq 100$ ).

T dòng tiếp theo, mỗi dòng gồm nhiều chuỗi các kí tự, có tổng độ dài không vượt quá  $10^5$ .

Các kí tự là chữ cái in hoa hoặc chữ cái thường.

## Output

Với mỗi test, in ra trên dòng số các giá trị mật mã của xâu, và xâu đó sau khi được mã hóa.

## Example

### Input:

```
3
Posts and Telecommunications Institute of Technology
i am a man
google
```

### Output:

```
41 postandelcmuifhgy
5 iamn
4 gole
```

```
1. using namespace std;
2. void xuli(string s){
3.     int n=s.length();
4.     int a[30];
5.     for(int i=0;i<26;i++) a[i]=0;
6.     for(int j=0;j<n;j++)
7.         for(int i=0;i<26;i++)
8.             if(s[j]==i+'a' || s[j]==i+'A') a[i]++;
9.     string s1;
10.    int d=0, t=0;
11.    for(int j=0;j<n;j++)
12.        for(int i=0;i<26;i++){
13.            if((s[j]==i+'a' || s[j]==i+'A') && a[i]>=2)
14.                t+=a[i];
15.            if(s[j]==i+'a' && a[i]>0){
16.                s1[d]=s[j];
17.                d++;
18.                a[i]=0;
19.            }
20.            if(s[j]==i+'A' && a[i]>0){
21.                s1[d]=s[j]+32;
22.                d++;
23.                a[i]=0;
24.            }
25.        }
26.    }
```

```

25.         cout<<t<<" ";
26.         for(int i=0;i<d;i++) cout<<s1[i];
27.         cout<<"\n";
28.     }
29.     main() {
30.         int n;
31.         string *s;
32.         cin>>n;
33.         s=new string[n+1];
34.         for(int i=0;i<=n;i++) {
35.             getline(cin,s[i]);
36.         }
37.         for(int i=1;i<=n;i++) xuli(s[i]);
38.     }
39.

```

## 87. Về quê

Cooper đang về quê, anh ra bến xe Blabla để lên xe khách về quê, xe khách có 34 ghế ngồi: 4 ghế ở cuối và 3 ghế ở những hàng còn lại (2 ghế ở bên trái ở giữa là đường đi) . Để dễ dàng xếp khách, các nhà xe xếp khách mới lên xe vào ghế trống bên trái nhất của hàng cuối cùng.

Số hành khách trên xe được ghi ở bảng điện tử bên ngoài xe, Cooper muốn ngồi ở vị trí ưa thích, anh biết cách xếp của các nhà xe vì vậy anh muốn phác họa lại các vị trí trên xe khách để xem mình có được ngồi vị trí ưa thích không.

Các bạn hãy giúp anh ấy.

### Input

Chứa số tự nhiên  $n$  là số khách hàng đã lên xe ( $0 \leq n \leq 34$ ).

### Output

Chứa bản phác họa xe khách như trong test ví dụ. Kí tự '#' thể hiện ghế chưa có người ngồi, 'O' thể hiện ghế đã có người ngồi, 'D' là bác tài xế, các kí tự còn lại thể hiện xe khách.

### Example

#### Input:

20

#### Output:

```

+-----+
|0.0.0.0.0.0.0.#.#.#.#.|D|)
|0.0.0.0.0.0.0.#.#.#.#.#|. |
|0.....|

```

|0.0.0.0.0.0.0.#.#.#.#.#.|.)

+-----+

```
1. public class Main {
2.
3.     static void solve(char a[][]) {
4.         String dau = "+-----+";
5.         System.out.println(dau);
6.         for (int i = 0; i < 4; i++) {
7.             System.out.print("|");
8.             for (int j = 0; j < a[i].length; j++)
9.                 System.out.print(a[i][j] + ".");
10.            switch (i) {
11.                case 0:
12.                    System.out.println("|D|");
13.                    break;
14.                case 1:
15.                    System.out.println("|.|");
16.                    break;
17.                case 2:
18.                    System.out.println("..|");
19.                    break;
20.                case 3:
21.                    System.out.println("|.|");
22.                    break;
23.            }
24.        }
25.        System.out.println(dau);
26.    }
27.
28.    public static void main(String[] args) {
29.        Scanner sc = new Scanner(System.in);
30.        int n = sc.nextInt();
31.        char a[][] = new char[4][11];
32.        for (int i = 0; i < 11; i++)
33.            for (int j = 0; j < 4; j++) {
34.                if (j == 2) {
35.                    if (i == 0) {
36.                        if (n > 0) {
37.                            a[j][i] = 'O';
38.                            n--;
39.                        } else a[j][i] = '#';
40.                    } else a[j][i] = '.';
41.                } else {
42.                    if (n > 0) {
43.                        a[j][i] = 'O';
44.                        n--;
45.                    } else a[j][i] = '#';
46.                }
47.            }
48.        solve(a);
    }
```

```
49.     }
50. }
51.
```

### 88. Cùng bet với EarthShaker

EarthShaker là một con bò huyền thoại với cái chày to vô đối, hẳn có tài dùng cái chày này gây ra động đất và có thể giết hết những kẻ thù ở trong một bán kính nào đó. Nhưng dù gì thì EarthShaker cũng chỉ là một con bò và anh thường đập rất lung tung, chính vì thế các bet thủ đã tạo ra một đấu trường rồi thả creeps và EarthShaker vào trong đó. Đấu trường có diện tích là S và có X con creeps trên một đơn vị diện tích, có tất cả 5 bet thủ cùng dự đoán số creeps sẽ bị tiêu diệt.

Bạn hãy tính xem số chênh lệch của các bet thủ là bao nhiêu.

#### Input

Dòng đầu tiên là 2 số X và S cách nhau bởi dấu cách ( $1 \leq X \leq 10$ ,  $1 \leq S \leq 1000$ ).

Dòng thứ 2 chứa 5 số nguyên a, b, c, d, e lần lượt là dự đoán của 5 bet thủ (a, b, c, d, e < 1000000).

#### Output

In ra 5 số là số chênh lệch của các bet thủ.

#### Example

##### Input:

5 20

99 101 1000 0 97

##### Output:

-1 1 900 -100 -3

```
1. class Defeat {
2.     //Try hard 3-5 accepts a day to defeat Cuy and get a cup of
   coffee
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         int x = sc.nextInt(), y = sc.nextInt(), a;
7.         for(int i=0; i<5; i++) {
8.             a = sc.nextInt();
9.             System.out.print(a-x*y + " ");
10.        }
11.
12.    }
13.
```

```
14. }
```

### 89. Điện tử số

Trong giờ thí nghiệm môn Điện tử số, 0xb0b0 phải thiết kế mạch để hiển thị 1 số có 2 chữ số trên 2 đèn LED 7 đoạn. Tuy nhiên, do lười nên anh ta không thiết kế được chuẩn cho lắm. Số anh ta thu được thiếu một vài nét so với số ban đầu.

Hãy giúp 0xb0b0 đếm xem có bao nhiêu số có thể khôi phục được nếu thêm 1 vài nét hoặc không thêm vào số hiện tại của anh ta.



Input

Một số  $n$  gồm 2 chữ số ( $0 \leq n \leq 99$ ) là số hiện tại của 0xb0b0 trên bảng mạch. Nếu  $n < 10$  thì ta nhập dưới dạng 0n.

Output

In ra một số nguyên duy nhất là đáp số của bài toán.

Example

Test 1:

Input:

89

Output:

2

Test 2:

Input:

00

Output:

4

```
1. class Defeat {
2.     //Try hard 3-5 accepts a day to defeat Cuy and get a cup of
   coffee
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
```

```

6.         String s = sc.next();
7.         System.out.println(value(s.charAt(0)) *
    value(s.charAt(1)));
8.     }
9.
10.    private static int value(char c) {
11.        switch (c) {
12.            case '0':
13.                return 2;
14.            case '1':
15.                return 7;
16.            case '2':
17.                return 2;
18.            case '3':
19.            case '4':
20.                return 3;
21.            case '5':
22.                return 4;
23.            case '6':
24.                return 2;
25.            case '7':
26.                return 5;
27.            case '8':
28.                return 1;
29.            case '9':
30.                return 2;
31.        }
32.        return 0;
33.    }
34.
35. }
36.

```

## 90. Tặng quà

Vào ngày cuối cùng của năm học lớp của Tí tổ chức tặng quà cho các bạn trong cùng một lớp. Lớp Tí có  $n$  người, mỗi người sẽ chuẩn bị một món quà. Sau đó mọi người trong lớp sẽ đổi quà cho nhau, biết sau khi đổi mỗi người sẽ chỉ có một món quà trong tay. Biết rằng người bạn thứ  $i$  sẽ tặng quà cho bạn  $p[i]$ , và mỗi bạn sẽ nhận được duy nhất một món quà. Đôi khi cũng có bạn nào đó chẳng đổi quà cho ai cả.

Giờ Tí muốn biết người thứ  $i$  sẽ được ai được tặng quà. Các bạn giúp Tí nhé!

### Input

Dòng đầu tiên chứa số nguyên  $n$  ( $1 \leq n \leq 100$ ) – số người trong lớp của Tí.

Dòng thứ 2 chứa  $n$  số nguyên là giá trị  $p[1], p[2], \dots, p[n]$  ( $1 \leq i \leq n, 1 \leq p[i] \leq n$ ).

### Output

In ra một dòng chứa  $n$  số lần lượt là số thứ tự của người đã tặng quà cho người thứ  $i$ .

Example

**Input:**

4

2 3 4 1

**Output:**

4 1 2 3

```
1. class Defeat {
2.     //Try hard 3-5 accepts a day to defeat Cuy and get a cup of
   coffee
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         int n = sc.nextInt(), a[] = new int[n + 1];
7.         for (int i = 1; i <= n; i++) a[sc.nextInt()] = i;
8.         for (int i = 1; i <= n; i++) System.out.print(a[i] + "
   ");
9.     }
10.
11. }
```

## 91. Tiệc bia

Sau khi trao quà tại lớp, Tí cùng cả lớp đi ăn nhậu. Sau khi đến quán bia, lớp của Tí gọi ra  $n$  chai nhưng khổ là lại không có gì để mở. May thay, có một số chai bia hãng này (có thể dùng chai nguyên hoặc nắp chai đã mở) có thể dùng để mở nắp chai bia hãng khác.

Giờ các bạn giúp Tí tìm xem có bao nhiêu chai bia mà không thể mở được bằng các chai hãng khác.

Input

Dòng đầu tiên là số nguyên  $n$  ( $1 \leq n \leq 100$ ) – số lượng chai bia gọi ra.

$n$  dòng tiếp theo, dòng thứ  $i$  chứa 2 số nguyên  $a[i]$ ,  $b[i]$ , với  $a[i]$  là hãng của chai bia  $i$  và chai bia này có thể dùng để mở nắp cho chai bia có hãng là  $b[i]$  ( $1 \leq a[i], b[i] \leq 1000$ ). Nếu  $a[i] = b[i]$ , có nghĩa chai bia thứ  $i$  có thể được mở bằng một chai bia khác cùng hãng.

Output

In ra một số nguyên là số lượng chai bia không thể được mở.

Example

Test 1:

Input:

4

1 1  
2 2  
3 3  
4 4

output:

4

```
1. using namespace std;
2. int n,*a,*b;
3. bool bia(int i){
4.     for(int j=0;j<n;j++){
5.         if(j!=i&& a[i]==b[j]) return true;
6.     }
7.     return false;
8. }
9.
10. main(){
11.     cin>>n;
12.     a=new int[n+1];
13.     b=new int[n+1];
14.     int m=n;
15.     for(int i=0;i<n;i++) cin>>a[i]>>b[i];
16.     for(int i=0;i<n;i++) if(bia(i)) m--;
17.     cout<<m;
18. }
19.
```

## 92. Phép dịch

Cho trước dãy ký tự:

ABCDEFGHIJKLMNOPQRSTUVWXYZ\_.

Phép dịch K trong dãy này được định nghĩa là đẩy một ký tự đi K vị trí.

Ví dụ: phép dịch 1 là 'A'→'B', 'B'→'C', ..., 'Z'→'\_', '\_'→'.', và '.'→'A'.

Phép dịch 3 là: 'A'→'D', 'B'→'E', ..., '.'→'C'.

Bài toán đặt ra là cho trước số nguyên K và một xâu ký tự, hãy ghi ra kết quả phép dịch K tương ứng của xâu đó sau khi đã đảo ngược thứ tự các chữ cái.

Input

Mỗi bộ test ghi trên một dòng số nguyên  $1 \leq N \leq 27$ , tiếp theo là khoảng trống rồi đến xâu S (không quá 40 ký tự và chỉ bao gồm các chữ cái in hoa).



Bộ test cuối cùng có một số 0.

Output

Với mỗi bộ test, ghi ra kết quả phép dịch sau khi đã đảo ngược thứ tự các chữ cái.

Example

**Input:**

```
1 ABCD
3 YO_THERE.
1 .DOT
14 ROAD
9 SHIFTING_AND_ROTATING_IS_NOT_ENCRYPTING
2 STRING_TO_BE_CONVERTED
1 SNQZDRQDUDQ
0
```

**Output:**

```
EDCB
CHUHKWBR.
UPEA
ROAD
PWRAYF_LWNHAXWH.RHPWRAJAX_HMWJHPWRAORQ.
FGVTGXPQEAGDAQVAIPKTVU
REVERSE_ROT
```

```
1. class Defeat {
2.     //Try hard 3-5 accepts a day to defeat Cuy and get a cup of
   coffee
3.
4.     static String PATTERN="ABCDEFGHIJKLMNOPQRSTUVWXYZ.";
5.     static int N=PATTERN.length();
6.
7.     public static void main(String[] args) {
8.         Scanner sc = new Scanner(System.in);
9.         while(true){
10.             int n=sc.nextInt();
11.             if(n==0) break;
12.             solve(n,sc.next());
13.         }
14.     }
15. }
```

```

16.     private static int position(char c) {
17.         if(c>='A'&&c<='Z') return c-'A';
18.         if(c=='_') return 26;
19.         if(c=='.') return 27;
20.         return 0;
21.     }
22.
23.     private static void solve(int n, String s) {
24.         String r="";
25.         for (int i = 0; i < s.length(); i++)
26.             r+=PATTERN.charAt((position(s.charAt(i))+n)%N);
27.         System.out.println(new StringBuilder(r).reverse());
28.     }
29. }

```

### 93. Số may mắn

Ds

Để chọn ra con số may mắn của riêng CLB IT - PTIT, chủ nhiệm CLB yêu cầu mỗi thành viên lựa chọn một số nguyên dương bất kỳ trong khoảng từ 1 đến 1000. Sau đó con số nào được chọn bởi nhiều người nhất thì sẽ là số may mắn. Nếu có nhiều con số được chọn nhiều lần như nhau thì sẽ ưu tiên chọn con số nhỏ hơn.

Giả sử số thành viên của CLB cũng không thể quá 1000 người. Hãy giúp CLB chọn ra số may mắn.

Input

Dòng đầu tiên ghi số bộ test (không quá 100). Dòng đầu của mỗi bộ test ghi số N, là tổng số thành viên của CLB.

Tiếp theo là N dòng, mỗi dòng ghi một giá trị được chọn.

Output

Với mỗi bộ test, ghi ra trên một dòng số may mắn tìm được.

Example

**Input:**

```

3
3
42
42
19
4
7

```

99

99

7

5

11

12

13

14

15

**Output:**

42

7

11

```
1. class Main {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         int testcase = sc.nextInt();
5.         while (testcase-- > 0) {
6.             int n = sc.nextInt(), a[] = new int[1001], max =
0, iMax = 0;
7.             for (int i = 1; i < 1001; i++)
8.                 a[i] = 0;
9.             for (int i = 0; i < n; i++)
10.                 a[sc.nextInt()]++;
11.             for (int i = 1; i < 1001; i++)
12.                 if (a[i] > max) {
13.                     iMax = i;
14.                     max = a[i];
15.                 }
16.             System.out.println(iMax);
17.         }
18.     }
19. }
20.
```

94. Tìm chữ số còn thiếu

Đề bài:

Cho một biểu thức toán học với các phép toán cơ bản cộng (+), trừ (-) và nhân (\*) trong đó có một số vị trí bị ẩn đi bằng dấu ?. Giả sử tất cả các vị trí có dấu ? đều cùng là một chữ số. Hãy tìm ra chữ số bị ẩn dấu.

Một số chú ý:

- Chữ số đầu tiên của các số hạng luôn khác 0 (trừ khi số đó chính là 0)
- Giá trị các số hạng luôn nằm trong khoảng -999999 đến 999999.
- Các vị trí có dấu ? đều là một chữ số cố định, và chữ số này phải khác với các chữ số đã nhìn thấy trong phép toán.

Input

Dòng đầu ghi số bộ test (không quá 100).

Mỗi bộ test ghi ra trên một dòng phép toán theo định dạng:

`[number][op][number]=[number]`

Luôn có ít nhất một dấu ? trong biểu thức. Không có khoảng trống hay các ký tự khác.

Output

Với mỗi bộ test, ghi ra giá trị chữ số tìm được ứng với dấu ?. Nếu không thể có chữ số nào thỏa mãn thì ghi ra số -1.

Nếu có nhiều đáp án, hãy in ra chữ số nhỏ nhất thỏa mãn.

Example

**Input :**

5

1+1=?

123\*45?=5?088

-5?\* -1=5?

19- -45=5?

?\*?\*?=302?

**Output :**

2

6

0

-1

5

```

1. class Main2 {
2.
3.     static boolean op(char c) {
4.         return c == '+' || c == '-' || c == '*' || c == '/';
5.     }
6.
7.     public static void main(String[] args) {
8.         Scanner sc = new Scanner(System.in);
9.         int n = Integer.parseInt(sc.nextLine());
10.        while (n-- > 0)
11.            solve(sc.nextLine());
12.    }
13.
14.    private static void solve(String s) {
15.        int opIndex = 1, equIndex = 1;
16.        for (int i = 1; i < s.length() - 1; i++)
17.            if ((op(s.charAt(i)) && !op(s.charAt(i +
18.                1))) || (op(s.charAt(i)) &&
19.                    s.charAt(i + 1) == '-')) {
20.                opIndex = i;
21.                break;
22.            }
23.        for (int i = opIndex; i < s.length() - 1; i++)
24.            if (s.charAt(i) == '=') {
25.                equIndex = i;
26.                break;
27.            }
28.        // System.out.println(opIndex + " - " +
29.            equIndex);
30.        String a = s.substring(0, opIndex);
31.        String b = s.substring(opIndex + 1, equIndex);
32.        String c = s.substring(equIndex + 1);
33.        // System.out.println(a + " " + b + " " + c);
34.
35.        String a1 = a.replaceAll("\\?", "0");
36.        String b1 = b.replaceAll("\\?", "0");
37.        String c1 = c.replaceAll("\\?", "0");
38.
39.        int a2 = Integer.parseInt(a1);
40.        int b2 = Integer.parseInt(b1);
41.        int c2 = Integer.parseInt(c1);
42.
43.        if ((a2 + "").length() == a1.length()
44.            && (b2 + "").length() == b1.length()
45.            && (c2 + "").length() ==
46.                c1.length())) {
47.            int zero = 0;
48.            // System.out.println(s.charAt(opIndex));

```

```

46.             switch (s.charAt(opIndex)) {
47.             case '+':
48.                 if (a2 + b2 == c2 && !a.contains("")
+ zero)
49.                     && !b.contains("") +
zero) && !c.contains("") + zero)) {
50.                     System.out.println(zero);
51.                     return;
52.                 }
53.                 break;
54.             case '-':
55.                 if (a2 - b2 == c2 && !a.contains("")
+ zero)
56.                     && !b.contains("") +
zero) && !c.contains("") + zero)) {
57.                     System.out.println(zero);
58.                     return;
59.                 }
60.                 break;
61.             case '*':
62.                 if (a2 * b2 == c2 && !a.contains("")
+ zero)
63.                     && !b.contains("") +
zero) && !c.contains("") + zero)) {
64.                     System.out.println(zero);
65.                     return;
66.                 }
67.                 break;
68.             }
69.         }
70.
71.         for (int i = 1; i <= 9; i++) {
72.             a1 = a.replaceAll("\\?", "" + i);
73.             b1 = b.replaceAll("\\?", "" + i);
74.             c1 = c.replaceAll("\\?", "" + i);
75.
76.             a2 = Integer.parseInt(a1);
77.             b2 = Integer.parseInt(b1);
78.             c2 = Integer.parseInt(c1);
79.
80.             switch (s.charAt(opIndex)) {
81.             case '+':
82.                 if (a2 + b2 == c2 && !a.contains("")
+ i) && !b.contains("") + i)
83.                     && !c.contains("") +
i)) {
84.                     System.out.println(i);
85.                     return;
86.                 }
87.                 break;
88.             case '-':

```

```

89.         if (a2 - b2 == c2 && !a.contains("")
+ i) && !b.contains("") + i)
90.             && !c.contains("") +
i)) {
91.                 System.out.println(i);
92.                 return;
93.             }
94.             break;
95.         case '*':
96.             if (a2 * b2 == c2 && !a.contains("")
+ i) && !b.contains("") + i)
97.                 && !c.contains("") +
i)) {
98.                     System.out.println(i);
99.                     return;
100.                 }
101.             }
102.         }
103.     }
104.     System.out.println(-1);
105. }
106. }

```

95. Chọn chỗ

Đề:

N'aix the Lifestealer sau khi về quê nghỉ Tết xong thì chuẩn bị bắt xe để lên Hà Nội học đại học tiếp. Nhưng anh lại là 1 thanh niên bủn xỉn, chỉ muốn đi xe chùa mà không muốn trả tiền, nhờ kỹ năng có một không hai của mình mà anh có thể bắt xe mà không mất tiền.

Đoàn xe mà N'aix muốn bắt có n chỗ ngồi, kỹ năng của anh là tìm ra chỗ trống có chỉ số nhỏ nhất trong đoàn xe đầy để ngồi. Hãy giúp N'aix tìm được chỗ ngồi hợp lý.

Input

Dòng đầu tiên chứa số n (  $1 \leq n \leq 3000$  ).

Dòng sau chứa n số nguyên dương x là chỗ đã có người ngồi (  $1 \leq x \leq 3000$  ).

Output

In ra vị trí nhỏ nhất là chỗ mà N'aix sẽ ngồi (nếu đoàn xe đã chật thì in ra n + 1).

Example

**Input:**

3

1 2 4

**Output:**

```

1. class Defeat {
2.     //Try hard 3-5 accepts a day to defeat Cuy and get a cup of
   coffee
3.
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         int n = sc.nextInt(), min = Integer.MAX_VALUE, max = 0;
7.         boolean b = false;
8.         ArrayList<Integer> a = new ArrayList<Integer>();
9.         while (n-- > 0) {
10.             int x = sc.nextInt();
11.             a.add(x);
12.             if (x > max) max = x;
13.         }
14.         if(!a.contains(1)){
15.             System.out.println(1);
16.             return;
17.         }
18.         for (int i : a)
19.             if (i + 1 < min && !a.contains(i + 1)) {
20.                 min = i + 1;
21.                 b = true;
22.             }
23.         if (!b) min = max + 1;
24.         System.out.println(min);
25.     }
26.
27.
28. }

```

## 96. Bữa tiệc sinh nhật

Đề:

Để chuẩn bị cho tiệc cưới của con gái thứ 35 của mình - tiểu thư Charlotte Pudding và con trai thứ 3 của gia tộc Vinsmock – Hắc cước Sanji, Charlotte Linlin hay còn được biết đến với cái tên Big Mom (1 trong 4 tử hoàng) đã cho chuẩn bị một chiếc bánh cưới rất đặc biệt.

Chiếc bánh có kích thước rất lớn, và có hình lục giác, với mỗi góc của chiếc bánh đúng bằng 120 độ. Để chiêu đãi khách, Big Mom đã cho cắt chiếc bánh ra, các lát cắt phải song song với cạnh của chiếc bánh và cách nhau đúng 1 decimet. Vì thế ta được N chiếc bánh con mỗi chiếc bánh là 1 hình tam giác đều có cạnh đúng bằng 1 decimet.

Vậy có tất cả bao nhiêu chiếc bánh :/

Input

Dữ liệu vào trên đúng 1 dòng gồm 6 số  $a_1, a_2, a_3, a_4, a_5, a_6$  cách nhau bởi dấu cách ( $1 \leq a_i \leq 1000$ ) là độ dài các cạnh của chiếc bánh hình lục giác (theo đơn vị decimet) theo đúng



thứ tự chiều kim đồng hồ. Dữ liệu vào đảm bảo rằng chiếc bánh là hình lục giác có các góc bằng 120 độ.

Output

In ra một số nguyên N duy nhất là số bánh hình tam giác đều có kích thước 3 cạnh đúng bằng 1.

Example

Test 1:

Input:

1 1 1 1 1 1

Output:

6

```
1. class Main2 {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         int a[]=new int[6];
5.         for(int i=0;i<6;i++)a[i]=sc.nextInt();
6.         System.out.println((a[1] + a[2] + a[3])*(a[1] + a[2] +
   a[3])
7.                             -a[1]*a[1]-a[3]*a[3] -a[5]*a[5] );
8.     }
9.
10. }
```

97. Vi khuẩn

Đề:

Tyrion nuôi cấy vi khuẩn trong 1 ống nghiệm. Ban đầu ống nghiệm rỗng. Mỗi sáng Tyrion có thể một lượng bất kỳ (có thể bằng 0) vi khuẩn vào ống nghiệm. Mỗi đêm, mỗi vi khuẩn trong ống sẽ tăng lên gấp đôi. Một ngày, Tyrion mong muốn nhìn thấy chính xác x vi khuẩn.

Hãy tính xem số lượng vi khuẩn tối thiểu Tyrion cần phải bỏ vào.

Input

Số nguyên x (  $1 \leq x \leq 1\,000\,000\,000$  ).

Output

In ra duy nhất 1 số là đáp án của bài toán.

Example

**Input:**

5

**Output:**

2

Giải thích:

Ngày thứ nhất Tyrion bỏ 1 vi khuẩn. Đến buổi sáng thứ 3, ông ta có 4 con vi khuẩn trong ống nghiệm. Ông ta bỏ thêm 1 con nữa là sẽ có 5.

```
1. main() {
2.     long long n = 1, count = 1;
3.     cin >> n;
4.
5.     while (n > 1) {
6.         if (n % 2 == 1) {
7.             count++;
8.         }
9.         n /= 2;
10.    }
11.    cout << count;
12. }
```

98. Alex gửi tiền ngân hàng

Đề:

```
1. class Main {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         int n = sc.nextInt();
5.         if (n >= 0)
6.             System.out.println(n);
7.         else {
8.             n = Math.abs(n);
9.             int m1 = n / 10, m2 = n / 100 * 10 + n %
10.                10;
11.             System.out.println(Math.max(-m1, -
12.                m2));
13.         }
14.     }
```

### 99. Cơ số 3

Đề:

```
1. class Test {
2.     public static String convert3(int n) {
3.         String s = "";
4.         while (n > 0) {
5.             s += (n % 3);
6.             n /= 3;
7.         }
8.         return new
           StringBuilder(s).reverse().toString();
9.     }
10.
11.     public static String tor(String x, String y)
12.     {
13.         int n = x.length(), m = y.length();
14.         if (m > n) {
15.             n = m;
16.             while (x.length() != m) {
17.                 x = "0" + x;
18.             }
19.         } else if (n > m) {
20.             m = n;
21.             while (y.length() != m) {
22.                 y = "0" + y;
23.             }
24.         }
25.         String t = "";
26.         for (int i = 0; i < n; i++) {
27.             char a = x.charAt(i), c =
               y.charAt(i);
28.             if (c >= a) {
29.                 t += (c - a);
30.             } else {
31.                 t += (c + 3 - a);
32.             }
33.         }
34.         // System.out.println("tor : " + t);
35.         return t;
36.     }
}
```

```

37.         public static int convert10(String n) {
38.             int t = 0, size = n.length();
39.             for (int i = 0; i < size; i++) {
40.                 t = t * 3 + (n.charAt(i) - '0');
41.                 //          System.out.println("i = " + i + "
t= " + t);
42.             }
43.             return t;
44.         }
45.
46.         public static void main(String[] args) {
47.             Scanner sc = new Scanner(System.in);
48.             int a = sc.nextInt(), c = sc.nextInt();
49.             String n = tor(convert3(a),
convert3(c));
50.             System.out.println(convert10(n));
51.         }
52.     }
53.

```

100. Kim tự tháp

Đề:

B có n khối hộp lập phương và anh ta quyết định xếp nó thành hình kim tự tháp. Kim tự tháp tầng trên cùng có 1 khối, tầng thứ 2 có  $1 + 2 = 3$  khối, tầng thứ 3 có  $1 + 2 + 3 = 6$  khối, cứ như vậy cho các đỉnh ở dưới. LB muốn xác định với n khối hộp, chiều cao tối đa của kim tự tháp là bao nhiêu?

## Input

Một dòng chứa số n ( $1 \leq n \leq 10^4$ ).

## Output

Đáp án của bài toán.

## Example

Test 1:

Input:

1

Output:

1

```
1. class Main {
2.
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int n = sc.nextInt(), s = 0, i, a[] = new
           int[2000];
6.         for (i = 1; a[i - 1] <= 10000; i++) {
7.             s += i;
8.             a[i] = a[i - 1] + s;
9.         }
10.        for (int j = 1; j < i - 1; j++)
11.            if (a[j] <= n && a[j + 1] > n) {
12.                System.out.println(j);
13.                break;
14.            }
15.        }
16.    }
17.
```

101. Số nguyên

Đề:

LB mới tìm hiểu về Java và thấy rằng Java có rất nhiều kiểu dữ liệu biểu diễn 1 số nguyên:

- Byte: các số thuộc đoạn [-128, 127]

- Short: các số thuộc đoạn [-32768, 32767]

- Int: các số thuộc đoạn [-2147483648, 2147483647]

- Long: các kiểu số liệu thuộc đoạn [-9223372036854775808, 9223372036854775808]

- Đặc biệt Java hỗ trợ các số lớn hơn Long các số này thuộc kiểu dữ liệu BigInteger.

LB tự hỏi 1 số nguyên sẽ thuộc kiểu dữ liệu bé nhất có thể nào?

## Input

Một dòng chứa số n, số n chứa tối đa 100 ký tự số và không có số 0 ở đầu.

## Output

In trên 1 dòng 1 trong các từ sau "byte, short, int, long, BigInteger" để biểu diễn kiểu dữ liệu nhỏ nhất mà có thể biểu diễn số n.

## Example

Input:

696969

Output:

int

[Submit solution!](#)

```
1.using namespace std;
2.
3.int compare(string x, string y){
4.    if(x.compare(y) == 0){
5.        return 0;
6.    }
7.    int n = x.size(), m = y.size();
8.    if(n > m){
9.        return 1;
10.    }else if(n < m){
11.        return -1;
12.    }else{
13.        for(int i = 0; i < n; i++){
14.            if(x[i] > y[i]){
15.                // cout<<"Vi tri: "<<i<<"\n";
16.                return 1;
17.            }else if(x[i] < y[i]){
18.                // cout<<"Vi tri: "<<i<<"\n";
19.                return -1;
20.            }
21.        }
22.    }
23. }
24.
25. string dataType(string s){
26.     string types[] = {"byte", "short", "int",
27. "long", "BigInteger"};
27.     bool neg = false;
28.     if(s[0]=='-'){
29.         neg = true;
```

```

30.         s.erase(0, 1);
31.     }
32.     int i = 0;
33.     if ((!neg && compare(s, "127") <= 0) ||
34.         (neg && compare(s, "128") <= 0)){
35.         return types[i];
36.     }
37.     i++;
38.     if ((!neg && compare(s, "32767") <= 0) ||
39.         (neg && compare(s, "32768") <= 0)){
40.         return types[i];
41.     }
42.     i++;
43.     if ((!neg && compare(s, "2147483647") <= 0)
44.         ||
45.         (neg && compare(s, "2147483648") <= 0))
46.     {
47.         return types[i];
48.     }
49.     i++;
50.     if ((!neg && compare(s,
51.         "9223372036854775807") <= 0) ||
52.         (neg && compare(s,
53.         "9223372036854775808") <= 0)){
54.         return types[i];
55.     }
56.     i++;
57.     return types[i];
58. }
59. main(){
60.     string s;
61.     cin>>s;
62.     cout<<dataType(s);
63. }

```

102. Thăng chức

Đề:

Steve Zob đã làm việc tại Microsoft được 5 năm và hiện vẫn đang làm lập trình viên tại đây. Để có thể được thăng chức lên Quản lý dự án, Steve phải vượt qua 1 bài test nhỏ sau:

Cho 2 dãy số có cùng chiều dài  $n$  là  $a[1], a[2], \dots, a[n]$  và  $b[1], b[2], \dots, b[n]$ . Chúng ta định nghĩa hàm  $o(x, l, r)$  là kết quả của phép bit OR tất cả các phần tử từ  $l$  đến  $r$  trong mảng  $x$ .

$o(x, l, r) = x[l] \mid x[l+1] \mid \dots \mid x[r-1] \mid x[r]$

Hãy xác định giá trị lớn nhất của  $O = o(a, l, r) + o(b, l, r)$  với mọi  $l, r$  thỏa mãn  $1 \leq l \leq r \leq n$ .

## Input

Dòng đầu tiên nhập số tự nhiên  $n$  ( $1 \leq n \leq 1000$ ) là chiều dài của 2 dãy số

Dòng thứ 2 nhập  $n$  số tự nhiên  $a[1], a[2], \dots, a[n]$  ( $1 \leq a[i] \leq 10^9$ )

Dòng thứ 3 nhập  $n$  số tự nhiên  $b[1], b[2], \dots, b[n]$  ( $1 \leq b[i] \leq 10^9$ )

## Output

In ra 1 số tự nhiên duy nhất là kết quả của bài toán

## Example

Test 1:

Input:

5

1 2 4 3 2

2 3 3 12 1

Output:

22

Test 2:

Input:

10

13 2 7 11 8 4 9 8 5 1

5 7 18 9 2 3 0 11 8 6

Output:



```

1. public class Main {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         int n = sc.nextInt();
5.         // long a[] = new long[n], b[] = new long[n];
6.         int a[] = new int[n], b[] = new int[n];
7.         for (int i = 0; i < n; i++)
8.             a[i] = sc.nextInt();
9.         for (int i = 0; i < n; i++)
10.            b[i] = sc.nextInt();
11.        // sc.close();
12.
13.        long max = 0;
14.        for (int i = 0; i < n; i++) {
15.            long aXor = (long)a[i], bXor
16.            = (long) b[i];
17.            max = Math.max(aXor + bXor, max);
18.            for (int j = i + 1; j < n; j++) {
19.                aXor |= (long) a[j];
20.                bXor |= (long) b[j];
21.                max = Math.max(aXor + bXor,
22.                max);
23.            }
24.        }
25.        System.out.println(max);
26.    }
27. }

```

103. Khóa vòng

Đề:

Bạn có 1 chiếc khóa vòng gồm nhiều vị trí. Mỗi vị trí gồm 1 chữ số có giá trị từ 0 đến 9. Có thể xoay ngược hoặc xuôi. Cho vị trí trước đó và hiện tại. Tính số lần xoay ít nhất để từ vị trí trước đó có được vị trí hiện tại?

## Input

Dòng đầu gồm 1 chữ số  $n$  ( $1 \leq n \leq 10000$ ) là số lượng vị trí trên khóa.

2 dòng tiếp theo gồm 2 số  $s$  và  $t$  có độ dài  $n$  (có thể bắt đầu bằng 0).

## Output

Đáp án của bài toán.

## Example

**Input:**

5

82195

64723

**Output:**

13

Giải thích:

Vị trí số 1: 8 → 7 → 6

Vị trí số 2: 2 → 3 → 4

Vị trí số 3: 1 → 0 → 9 → 8 → 7

Vị trí số 4: 9 → 0 → 1 → 2

Vị trí số 5: 5 → 4 → 3

```
1. class BaiG {
2.     public static void main(String[] args) {
3.         Scanner sc = new Scanner(System.in);
4.         int n = sc.nextInt(), t = 0;
5.         String x = sc.next(), y = sc.next();
6.         for (int i = 0; i < n; i++) {
7.             int a = x.charAt(i) - '0', b =
y.charAt(i) - '0';
8.             int p = Math.abs(a - b), q;
9.             if (a > b) {
10.                 q = 10 - a + b;
11.             } else {
12.                 q = 10 - b + a;
13.             }
14.             t += Math.min(p, q);
15.         }
16.         System.out.println(t);
17.     }
18. }
```

104. Cấp số cộng hay cấp số nhân

Đề:

Cho một dãy số có 4 số nguyên trong đó khuyết một vị trí chưa rõ giá trị. Bạn hãy kiểm tra xem dãy đó có thể là một cấp số cộng hoặc cấp số nhân hay không. Nếu đúng, hãy điền số phù hợp vào vị trí bị khuyết.

## Input

Có không quá 100 bộ test. Mỗi bộ test ghi trên một dòng 4 số nguyên. Trong đó, vị trí có giá trị -1 là vị trí khuyết và cần xác định giá trị. Ba vị trí còn lại là một số trong khoảng 1 đến 10000. Trừ vị trí bằng -1, các số còn lại tạo thành một dãy không giảm.

Dòng cuối cùng của input chứa 4 số -1.

## Output

Với mỗi bộ test, in ra trên một dòng giá trị (từ 1 đến 10000) có thể điền vào vị trí bị khuyết. Hoặc nếu không tìm được số nào thỏa mãn thì ghi số -1.

## Example

**Input:**

```
1 2 -1 4
2 4 8 -1
7 8 -1 21
5 -1 11 14
-1 2 4 6
-1 -1 -1 -1
```

**Output:**

```
3
16
-1
8
-1
```

```
1.int dienSo(double a, double b, double c, int viTri){
2.    switch(viTri){
3.        case 0:
```

```

4.         if((a + c) / 2 == b){
5.             return a - (b - a);
6.         }
7.         if(c / b == b / a){
8.             double t = a / (b / a);
9.             if((int) t == t){
10.                 return t;
11.             }
12.         }
13.         break;
14.     case 3:
15.         if((a + c) / 2 == b){
16.             return c + (b - a);
17.         }
18.         if(c / b == b / a){
19.             double t = c * (b / a);
20.             if((int) t == t){
21.                 return t;
22.             }
23.         }
24.         break;
25.     case 1:
26.         if((b - a) / 2 == c - b){
27.             return (a + (c - b));
28.         }
29.         if(sqrt(b / a) == c / b){
30.             double t = a * (c / b);
31.             if((int) t == t){
32.                 return t;
33.             }
34.         }
35.         break;
36.     case 2:
37.         if((c - b) / 2 == b - a){
38.             return (b + (b - a));
39.         }
40.         if(sqrt(c / b) == b / a){
41.             double t = b * (b / a);
42.             if((int) t == t){
43.                 return t;
44.             }
45.         }

```

```

46.             break;
47.         }
48.         return -1;
49.     }
50.
51.     main() {
52.         double a, b, c, d;
53.         while(true) {
54.             cin>>a>>b>>c>>d;
55.             if(a == b && b == c && c == d && d == -
1) {
56.                 break;
57.             }
58.             int t;
59.             if(a == -1) {
60.                 t = dienSo(b, c, d, 0);
61.             }else if(b == -1) {
62.                 t = dienSo(a, c, d, 1);
63.             }else if(c == -1) {
64.                 t = dienSo(a, b, d, 2);
65.             }else{
66.                 t = dienSo(a, b, c, 3);
67.             }
68.             if(t >= 1 && t <= 10000) {
69.                 cout<<t<<"\n";
70.             }else{
71.                 t = -1;
72.                 cout<<t<<"\n";
73.             }
74.         }
75.     }

```

105. ABC

Cho đẳng thức  $a + b = c$ , trong 3 số này có 1 số bị mờ đi một chữ số (được thay bằng dấu ?), hãy tìm chữ số đó.

## Input

Dòng đầu chứa một số nguyên không âm a;

Dòng thứ hai chứa một số nguyên không âm b;

Dòng thứ ba chứa một số nguyên không âm c;

Cả ba số đều không vượt quá  $10^6$ . Một trong ba số sẽ bị thay một vị trí bằng dấu ?.

## Output

Gồm một dòng chứa một số chữ số là kết quả tương ứng của bộ test trong dữ liệu vào.

## Example

Input:

128

?2

200

Output:

7

```
1.
   class Main2 {
2.       public static void main(String[] args) {
3.           Scanner sc = new Scanner(System.in);
4.           String a = sc.nextLine(), b = sc.nextLine(),
           c = sc.nextLine();
5.           if (c.contains("?")) {
6.               String r = (Integer.parseInt(a) +
           Integer.parseInt(b)) + "";
7.               int i = c.indexOf("?");
8.               System.out.println(r.charAt(i));
9.           } else if (a.contains("?")) {
10.              String r = (Integer.parseInt(c) -
           Integer.parseInt(b)) + "";
11.              int i = a.indexOf("?");
12.              System.out.println(r.charAt(i));
13.           } else {
14.              String r = (Integer.parseInt(c) -
           Integer.parseInt(a)) + "";
15.              int i = b.indexOf("?");
16.              System.out.println(r.charAt(i));
17.           }
18.       }
19.   }
```

20.

106. Phương án khuyến mại

Trong ngày đầu tiên phát hành các số điện thoại di động “0913.xxx.xxx”, với sáu chữ số cuối tạo thành 1 số có 6 chữ số (không có chữ số 0 ở đầu), công ty VINAPHONE dự định khuyến mại cho N khách hàng đăng ký trước nhất các số điện thoại loại 1, M khách hàng kế tiếp số điện thoại loại 2 và K khách hàng cuối cùng các số điện thoại loại 3. Các số điện thoại loại 1, loại 2 và loại 3 có tính chất sau:

- Số Loại 3: là các số điện thoại mà sáu số cuối cùng của nó tạo thành một số thuận nghịch có sáu chữ số. Ví dụ số : 0913.257752.
- Số Loại 2: là các số điện thoại Loại 3 có tổng sáu số cuối cùng của nó là một số chia hết cho 10. Ví dụ số : 0913.104401.
- Số Loại 1: là các số điện thoại Loại 2 có sáu số cuối cùng của nó không chứa bất kỳ số 0 nào. Ví dụ số : 0913.686686.

Bài toán được đặt ra là cho trước một phương án N, M, K, hãy trả lời “YES” nếu công ty VINAPHONE thực hiện được, trả lời “NO” nếu công ty không thực hiện được.

**Input :** Dòng đầu tiên ghi số bộ test, không lớn hơn 100. Mỗi bộ test là một bộ 3 số N, M, K được ghi trên một dòng. Các số được ghi cách nhau một vài khoảng trống.

**Output:** Với mỗi bộ test, in ra màn hình trên một dòng giá trị “YES” hoặc “NO” tương ứng với phương án thực hiện được, hoặc phương án không thực hiện được.

## Example

**Input :**

```
5
100 100 200
50 150 200
100 50 300
120 50 500
140 50 700
```

**Output:**

```
NO
NO
YES
YES
NO
```

```

1.using namespace std;
2.main(){
3.    int test;
4.    cin>>test;
5.    int ba=0,hai=0,mot=0;
6.    for(long i=100000;i<1000000;i++){
7.        int m=0,n=i,tong=0,so0=0;
8.        while(n>0){
9.            m = m*10 + n%10;
10.           tong += n%10;
11.           if(n%10 == 0) so0++;
12.           n /= 10;
13.       }
14.       if(m == i){
15.           ba++;
16.           if(tong%10 == 0){
17.               hai++;
18.               if(so0 == 0) mot++;
19.           }
20.       }
21.   }
22.   //cout<<ba<<" "<<hai<<" "<<mot;
23.   while(test--){
24.       int l3,l2,l1;
25.       cin>>l1>>l2>>l3;
26.       if(l1+l2+l3 <= ba&&l1+l2 <= hai&&l1 <=
mot) cout<<"YES\n";
27.       else cout<<"NO\n";
28.   }
29. }
30.

```

107. Dãy con tăng dần tự nhiên bậc K

Đề:

Cho dãy gồm  $N$  số phân biệt  $A_N = \{a_1, a_2, \dots, a_N\}$  và số tự nhiên  $K (K \leq N \leq 100)$ . Ta gọi một *dãy con tăng dần bậc K* của dãy số  $A_N$  là một dãy các số gồm  $K$  phần tử trong dãy đó thỏa mãn tính chất tăng dần. Bài toán được đặt ra là hãy tìm số các *dãy con tăng dần bậc K* của dãy số  $A_N$ .



**Input:** Dòng đầu tiên ghi số bộ test, không lớn hơn 100. Mỗi bộ test được xây dựng theo khuôn dạng sau:

- Dòng đầu tiên ghi lại hai số  $N$  và  $K$  tương ứng với số phần tử của dãy số và bậc của dãy con.
- Dòng kế tiếp ghi lại  $N$  số của dãy số  $A_N$ , các số trong dãy không lớn hơn 100.

**Output:** Với mỗi bộ test, in ra màn hình số các dãy con tăng dần tự nhiên bậc  $K$  của dãy số  $A_N$ .

## Example

**Input:**

```
2
5 3
2 5 15 10 20
5 3
2 20 10 15 5
```

**Output:**

```
7
1
```

```
1.class dayback {
2.    public static int x[] = new int[101], dem;
3.    public static void xuly(int n, int k, int a[]){
4.        dem = 0; a[0] = 0;
5.        for(int i = 1; i <= k; i++)
6.            x[i] = i;
7.        quaylui(1, n, k, a);
8.        System.out.println(dem);
9.    }
10.    public static void quaylui(int i, int n, int
    k, int a[]){
11.        for(int j = x[i - 1] + 1; j <= n
    - k + i; j++){
12.            x[i] = j;
13.            if(i == k && a[x[k]] > a[x[k
    - 1]]) dem++;
14.            else if( a[x[i - 1]] <
    a[x[i]] ) quaylui(i + 1, n, k, a);
```

```

15.         }
16.     }
17.     public static void main(String[] args) {
18.         Scanner nhap = new
Scanner(System.in);
19.         int sotest = nhap.nextInt();
20.         int n[] = new int [101], k[] =
new int [101];
21.         int a[][] = new int [101][101];
22.         for(int i = 0; i < sotest; i++){
23.             n[i] = nhap.nextInt();
24.             k[i] = nhap.nextInt();
25.             for(int j = 1; j <= n[i]; j+
+)
26.                 a[i][j] =
nhap.nextInt();
27.         }
28.         for(int i = 0; i < sotest; i++)
29.             xuly(n[i], k[i], a[i]);
30.     }
31. }
32.

```

108. Số may mắn

Đề:

John rất thích con số 86 vì theo John đó là con số may mắn. Khi bắt gặp một số nguyên X John muốn kiểm tra xem trong dạng biểu diễn của nó kết thúc là số 86 hay không? Ví dụ số 111539786 kết thúc là số 86, còn số 123456789 thì không. Nhiệm vụ của bạn là viết một chương trình đọc số nguyên X và kiểm tra xem trong dạng biểu diễn của nó kết thúc là số 86 hay không?

## Input

Dữ liệu vào gồm nhiều bộ dữ liệu tương ứng với nhiều test. Dòng đầu tiên chứa một số nguyên dương không lớn hơn 20 là số lượng các bộ dữ liệu. Các dòng tiếp theo chứa các bộ dữ liệu.

## Output

Với mỗi bộ dữ liệu, ghi ra trên một dòng câu trả lời, ghi số 1 nếu trong dạng biểu diễn của nó kết thúc là số 86, ghi số 0 trong trường hợp ngược lại.

## Example

**Input:**

```
3
111539786
123456789
8686
```

**Output:**

```
1
0
1
```

```
1.using namespace std;
2.int mm(long long n){
3.    int m = n%100;
4.    if(m == 86) return 1;
5.    return 0;
6.}
7.main(){
8.    int test;
9.    cin>>test;
10.    while(test--){
11.        long long n;
12.        cin>>n;
13.        cout<<mm(n)<<"\n";
14.    }
15.}
16.
```

109. ĐOán số

Đề:

An và Bình cùng chơi trò đoán số như sau: An nghĩ ra một số nguyên  $n$  trong đoạn  $[1,3]$  sau đó Bình sẽ đoán số mà An nghĩ bằng cách đặt các câu hỏi. Cụ thể, Bình sẽ nghĩ một số nguyên  $m$  và hỏi An: “Số cậu nghĩ có phải là số  $m$  hay không?”. Nếu  $m$  đúng là số mà An nghĩ thì An sẽ trả lời là “đúng”; còn nếu  $m$  lớn hơn số An nghĩ thì An sẽ trả lời là “lớn hơn”; An sẽ trả lời là “nhỏ hơn” nếu  $m$  nhỏ hơn  $n$ . Trò chơi kết thúc khi Bình đoán đúng được số mà An nghĩ.

Yêu cầu: Cho biết câu hỏi đầu tiên của Bình với  $m = 2$  và câu trả lời của An cho câu hỏi đó, hãy tìm số  $n$ .

## Input

Dữ liệu vào gồm nhiều bộ dữ liệu tương ứng với nhiều test. Dòng đầu tiên chứa một số nguyên dương không vượt quá 10 là số lượng các bộ dữ liệu. Các dòng tiếp theo chứa các bộ dữ liệu.

Mỗi bộ dữ liệu gồm một dòng duy nhất chứa một xâu là câu trả lời cho câu hỏi đầu tiên của Bình (xâu nhận giá trị là một trong ba xâu "dung", "lon hon", "nho hon" tương ứng là ba khả năng mà An trả lời "đúng", "lớn hơn", "nhỏ hơn").

## Output

Với mỗi bộ dữ liệu ghi ra trên một dòng, mỗi dòng một số nguyên là số  $n$  mà An nghĩ tương ứng với bộ dữ liệu trong dữ liệu vào.

## Example

Input:

1  
dung

Output:

2

```
1.
2. class Main2 {
3.     public static void main(String[] args) {
4.         Scanner sc = new Scanner(System.in);
5.         int n = sc.nextInt();
6.         while (n -- >= 0) {
7.             String s = sc.nextLine();
8.             if (s.equals("dung"))
9.                 System.out.println(2);
10.            else if (s.equals("lon hon"))
11.                System.out.println(1);
12.            else if (s.equals("nho hon"))
13.                System.out.println(3);
14.        }
15.    }
16.
17. }
18.
```

110. Ptit sumer

Đề:

Cho dãy số có N số nguyên dương và số M.

Yêu cầu bạn chọn ra 3 số nguyên sao cho tổng 3 số càng gần M càng tốt và không vượt quá M.

## Input

Dòng đầu tiên chứa số bộ test T. Sau đó là T bộ test, mỗi bộ test có dạng:

\*Dòng 1: Số tự nhiên N ( $1 \leq N \leq 100$ ) và số M ( $10 \leq M \leq 300\,000$ )

\*Dòng 2: Chứa N số nguyên dương (không quá 100 000) cách nhau bởi dấu cách.

Dữ liệu luôn tồn tại tổng 3 số không lớn hơn M

## Output

Mỗi bộ test in trên một dòng là tổng của 3 số thỏa mãn điều kiện.

## Example

**Input:**

```
2
5 21
5 6 7 8 9
10 500
93 181 245 214 315 36 185 138 216 295
```

**Output:**

```
21
497
```

```
1.using namespace std;
2.main(){
3.    int test;
4.    long kq[101];
5.    cin>>test;
6.    for(int t = 1; t <= test; t++){
7.        int n;
8.        long m, a[101];
9.        kq[t] = 0;
10.        cin>>n>>m;
11.        for(int i = 1; i <= n; i++) cin>>a[i];
```

```

12.         for(int i = 1; i < n - 1; i++)
13.             for(int j = i + 1; j < n; j++)
14.                 for(int k = j + 1; k <= n; k++)
15.                     if(a[i] + a[j] + a[k] >
16.                         kq[t] && a[i] + a[j] + a[k] <= m)
17.                         kq[t] = a[i] + a[j] +
18.                             a[k];
19.             }
20.         for(int t = 1; t <= test; t++)
21.             cout<<kq[t]<<"\n";

```

111. Mã hóa thư

Đề: Tí và Tèo gửi thư trao đổi với nhau về buổi đi dã ngoại. Nhưng sợ bị mẹ phát hiện, cả hai đã nghĩ ra việc mã hóa thư trước khi gửi. Họ mã hóa khá đơn giản như sau:

\* Bức thư gồm có N kí tự. Người gửi sẽ mã hóa theo ma trận gồm R hàng và C cột với  $R.C = N$  và  $R \leq C$ . Nếu có nhiều ma trận thỏa mãn thì sẽ chọn ma trận có nhiều hàng nhất.

\* Người gửi viết bức thư dưới dạng sắp xếp theo hàng. (Các chữ cái được xếp vào hết hàng 1, 2, ... n). Và sẽ gửi thư dưới dạng sắp xếp theo cột.

Ngay buổi tối, Tí gửi bức thư đầu tiên về kế hoạch đi chơi cho Tèo. Các bạn hãy giải mã bức thư giúp Tèo với!

## Input

Một dòng duy nhất là bức thư mà Tí gửi cho Tèo đã được mã hóa, là một xâu chứa các kí tự thường trong bảng chữ cái tiếng anh, không chứa dấu cách và có độ dài nhỏ hơn 100.

## Output

In ra bức thư sau khi đã được giải mã.

## Example

Test 1:

Input:

Bok

Output:

Bok

```
1. class Main2 {
2.     static void solve(String s) {
3.         int n = s.length(), r = (int) Math.sqrt(n),
           c = 0, d = 0;
4.         for (; r >= 1; r--)
5.             if (n % r == 0) {
6.                 c = n / r;
7.                 break;
8.             }
9.         char a[][] = new char[r][c];
10.        for (int i = 0; i < c; i++)
11.            for (int j = 0; j < r; j++)
12.                a[j][i] = s.charAt(d++);
13.        for (int i = 0; i < r; i++)
14.            for (int j = 0; j < c; j++)
15.                System.out.print(a[i][j]);
16.        System.out.println();
17.    }
18.
19.    public static void main(String[] args) {
20.        Scanner sc = new Scanner(System.in);
21.        solve(sc.nextLine());
22.    }
23. }
```