3-英文题试水

- 1720

Problem Description

Many classmates said to me that A+B is must needs.
If you can't AC this problem, you would invite me for night meal. ^_^

Input

Input may contain multiple test cases. Each case contains A and B in one line.
A, B are hexadecimal number.
Input terminates by EOF.

Output

Output A+B in decimal number in one line.

Sample Input

```
1 9
A B
a b
```

Sample Output

```
10
21
21
```

```c
#include <stdio.h>
int main()
{
    int a, b;
    while (~scanf("%x%x", &a, &b))
        printf("%d\n", a + b);
    return 0;
}
```

- 1062

Problem Description

Ignatius likes to write words in reverse way. Given a single line of text which is written by Ignatius, you should reverse all the words and then output them.

Input

The input contains several test cases. The first line of the input is a single integer T which is the number of test cases. T test cases follow.
Each test case contains a single line with several words. There will be at most 1000 characters in a line.

Output

For each test case, you should output the text which is processed.

Sample Input

```
3
olleh !dlrow
m'I morf .udh
I ekil .mca
```

Sample Output

```
hello world!
I'm from hdu.
I like acm.
```

Hint

Remember to use getchar() to read '\n' after the interger T, then you may use gets() to read a line and process it.

```cpp
#include <iostream>
#include <algorithm>
#include <string>
#include <vector>
using namespace std;
int main()
{
    int n;
    string str;
    cin >> n;
    getchar();
    while (n--)
    {
        getline(cin,str);
        vector<char> s(str.begin(),str.end());
        vector<char>::iterator start = s.begin();
        for (vector<char>::iterator it = s.begin() ;it!=s.end(); it++)
        {
            if (*it == ' ')
            {
                reverse(start, it);
                start = it + 1;
            }
            else if (it + 1 == s.end())
            {
                reverse(start, it+1);
                start = it + 1;
            }
        }
        for (vector<char>::iterator it = s.begin(); it != s.end(); it++)
            cout << *it;
        cout << endl;
    }
    return 0;
}
```

- 2104

Problem Description

The Children's Day has passed for some days .Has you remembered something happened at your childhood? I remembered I often played a game called hide handkerchief with my friends.
Now I introduce the game to you. Suppose there are N people played the game ,who sit on the ground forming a circle ,everyone owns a box behind them .Also there is a beautiful handkerchief hid in a box which is one of the boxes .
Then Haha(a friend of mine) is called to find the handkerchief. But he has a strange habit. Each time he will search the next box which is separated by M-1 boxes from the current box. For example, there are three boxes named A,B,C, and now Haha is at place of A. now he decide the M if equal to 2, so he will search A first, then he will search the C box, for C is separated by 2-1 = 1 box B from the current box A . Then he will search the box B ,then he will search the box A.
So after three times he establishes that he can find the beautiful handkerchief. Now I will give you N and M, can you tell me that Haha is able to find the handkerchief or not. If he can, you should tell me "YES", else tell me "POOR Haha".

Input

There will be several test cases; each case input contains two integers N and M, which satisfy the relationship: 1<=M<=100000000 and 3<=N<=100000000. When N=-1 and M=-1 means the end of input case, and you should not process the data.

Output

For each input case, you should only the result that Haha can find the handkerchief or not.

Sample Input

```
3 2
-1 -1
```

Sample Output

```
YES
```

```cpp
#include<iostream>
using namespace std;
int gcd(int a, int b)//辗转相除
{
    return b == 0 ? a : gcd(b, a % b);
}
int main()
{
    int m, n, t;
    while (cin >> m >> n && m >= 3 && n >= 1)
    {
        if (gcd(m, n) == 1)
            cout << "YES" << endl;
        else
            cout << "POOR Haha" << endl;
    }
}
//实际上就是判断2个数是否互质
//互质的2个数的最大公约数为1
//最小公倍数 = a*b/它们的最大公约数
```

- 1064

Problem Description

Larry graduated this year and finally has a job. He's making a lot of money, but somehow never seems to have enough. Larry has decided that he needs to grab hold of his financial portfolio and solve his financing problems. The first step is to figure out what's been going on with his money. Larry has his bank account statements and wants to see how much money he has. Help Larry by writing a program to take his closing balance from each of the past twelve months and calculate his average account balance.

Input

The input will be twelve lines. Each line will contain the closing balance of his bank account for a particular month. Each number will be positive and displayed to the penny. No dollar sign will be included.

Output

The output will be a single number, the average (mean) of the closing balances for the twelve months. It will be rounded to the nearest penny, preceded immediately by a dollar sign, and followed by the end-of-line. There will be no other spaces or characters in the output.

Sample Input

```
100.00
489.12
12454.12
1234.10
823.05
109.20
5.27
1542.25
839.18
83.99
1295.01
1.75
```

Sample Output

```
$1581.42
```

```c
#include <stdio.h>

int main()
{
    int cnt = 12;
    double n, sum=0;
    while (~scanf_s("%lf",&n))
        sum += n;
    printf("$%.2lf\n", sum / 12);
}
```

- 2734

Problem Description

A checksum is an algorithm that scans a packet of data and returns a single number. The idea is that if the packet is changed, the checksum will also change, so checksums are often used for detecting transmission errors, validating document contents, and in many other situations where it is necessary to detect undesirable changes in data.

For this problem, you will implement a checksum algorithm called Quicksum. A Quicksum packet allows only uppercase letters and spaces. It always begins and ends with an uppercase letter. Otherwise, spaces and letters can occur in any combination, including consecutive spaces.

A Quicksum is the sum of the products of each character's position in the packet times the character's value. A space has a value of zero, while letters have a value equal to their position in the alphabet. So, A=1, B=2, etc., through Z=26. Here are example Quicksum calculations for the packets "ACM" and "MID CENTRAL":

ACM: 1*1 + 2*3 + 3*13 = 46MID CENTRAL: 1*13 + 2*9 + 3*4 + 4*0 + 5*3 + 6*5 + 7*14 + 8*20 + 9*18 + 10*1 + 11*12 = 650

Input

The input consists of one or more packets followed by a line containing only # that signals the end of the input. Each packet is on a line by itself, does not begin or end with a space, and contains from 1 to 255 characters.

Output

For each packet, output its Quicksum on a separate line in the output.

Sample Input

```
ACM
MID CENTRAL
REGIONAL PROGRAMMING CONTEST
ACN
A C M
ABC
BBC
#
```

Sample Output

```
46
650
4690
49
75
14
15
```

```cpp
#include <iostream>
#include <string>
using namespace std;

int main(){

    string str;
    while(getline(cin, str)){
        if(str=="#")
            break;

        int result=0;
        for(int i=0; i<str.size(); i++){
            if(str[i]==' ')
                continue;
            result += (i+1)*(str[i]-'A'+1);
        }
        cout<<result<<endl;
    }
    return 0;
}
```

- 1170

Problem Description

The contest starts now! How excited it is to see balloons floating around. You, one of the best programmers in HDU, can get a very beautiful balloon if only you have solved the very very very... easy problem.
Give you an operator (+,-,*, / --denoting addition, subtraction, multiplication, division respectively) and two positive integers, your task is to output the result.
Is it very easy?
Come on, guy! PLMM will send you a beautiful Balloon right now!
Good Luck!

Input

Input contains multiple test cases. The first line of the input is a single integer T (0<T<1000) which is the number of test cases. T test cases follow. Each test case contains a char C (+,-,*, /) and two integers A and B(0<A,B<10000).Of course, we all know that A and B are operands and C is an operator.

Output

For each case, print the operation result. The result should be rounded to 2 decimal places If and only if it is not an integer.

Sample Input

```
4
+ 1 2
- 1 2
* 1 2
/ 1 2
```

Sample Output

```
3
-1
2
0.50
```

```cpp
#include<iostream>
#include<stdio.h>
using namespace std;
int main()
{
    int n,a,b;
    char s;
    cin>>n;
    while(n--)
    {
        double sum;
        cin>>s>>a>>b;
        switch(s)
        {
           case '+':sum=a+b;
                break;
           case '-':sum=a-b;
                break;
           case '*':sum=a*b;
                break;
           case '/':sum=(float)a/(float)b;
                break;
           default:
                break;
     }
     if(sum!=(int) sum)
        printf("%.2lf\n",sum);
     else
         printf("%.0lf\n",sum);
    }
    return 0;
}
```

- 1197

Problem Description

Find and list all four-digit numbers in decimal notation that have the property that the sum of its four digits equals the sum of its digits when represented in hexadecimal (base 16) notation and also equals the sum of its digits when represented in duodecimal (base 12) notation.

For example, the number 2991 has the sum of (decimal) digits 2+9+9+1 = 21. Since 2991 = 1*1728 + 8*144 + 9*12 + 3, its duodecimal representation is 1893(12), and these digits also sum up to 21. But in hexadecimal 2991 is BAF16, and 11+10+15 = 36, so 2991 should be rejected by your program.

The next number (2992), however, has digits that sum to 22 in all three representations (including BB016), so 2992 should be on the listed output. (We don't want decimal numbers with fewer than four digits - excluding leading zeroes - so that 2992 is the first correct answer.)

Input

There is no input for this problem.

Output

Your output is to be 2992 and all larger four-digit numbers that satisfy the requirements (in strictly increasing order), each on a separate line with no leading or trailing blanks, ending with a new-line character. There are to be no blank lines in the output. The first few lines of the output are shown below.

Sample Input

```
There is no input for this problem.
```

Sample Output

```
2992
2993
2994
2995
2996
2997
2998
2999
```

```cpp
#include <iostream>
#include<stdio.h>

using namespace std;

int main(void)
{
    int dec_sum=0,hex_sum=0,duo_sum=0;

    for(int i=1000; i<10000; i++)
    {
        //计算十进制各位的和
        while(i!=0)
        {
            dec_sum += i%10;
            dec_temp /= 10;
        }
        //计算16进制各位的和
        while(i!=0)
        {
            hex_sum += i%16;
            hex_temp /= 16;

        }
        //计算12进制各位的和
        while(i!=0)
        {
            duo_sum += i%12;
            duo_temp /= 12;
        }

        //三种方式的和都相等就输出，注意不能写成dec_sum==hex_sum==duo_sum
        if(dec_sum==hex_sum&&hex_sum==duo_sum)
```

```
            cout<<i<<endl;
        //在最后必须重新清零
        dec_sum=0;
        hex_sum=0;
        duo_sum=0;
    }
    return 0;
}
```

- 2629

Problem Description

Do you own an ID card?You must have a identity card number in your family's Household Register. From the ID card you can get specific personal information of everyone. The number has 18 bits,the first 17 bits contain special specially meanings:the first 6 bits represent the region you come from,then comes the next 8 bits which stand for your birthday.What do other 4 bits represent?You can Baidu or Google it.
Here is the codes which represent the region you are in.

Zhejiang 330000	Beijing 110000	Taiwan 710000	Hong Kong 810000
Macao 820000	Tibet 540000	Liaoning 210000	Shanghai 310000

However,in your card,maybe only 33 appears,0000 is replaced by other numbers.
Here is Samuel's ID number 331004198910120036 can you tell where he is from?The first 2 numbers tell that he is from Zhengjiang Province,number 19891012 is his birthday date (yy/mm/dd).

Input

Input will contain 2 parts:
A number n in the first line,n here means there is n test cases. For each of the test cases,there is a string of the ID card number.

Output

Based on the table output where he is from and when is his birthday. The format you can refer to the Sample Output.

Sample Input

```
1
330000198910120036
```

Sample Output

```
He/She is from Zhejiang,and his/her birthday is on 10,12,1989 based on the table.
```

```
#include<iostream>
#include<string>
using namespace std;
int main()
{
    int n,m;
    cin>>n;
    while(n--)
    {
        string str;
        string str1,year,mon,day;
        string des;

        cin>>str;
        str1=str.substr(0,2);//截取地址信息
        year=str.substr(6,4);//生日年份信息
        mon=str.substr(10,2);//生日月份信息
        day=str.substr(12,2);//生日信息

        int n=(str1[0]-'0')*10+str1[1]-'0';
        switch(n)
        {
          case 33: des="Zhejiang";break;
```

```
            case 11: des="Beijing";break;
            case 71: des="Taiwan";break;
            case 81: des="Hong Kong";break;
            case 82: des="Macao";break;
            case 54: des="Tibet";break;
            case 21: des="Liaoning";break;
            case 31: des="Shanghai";break;
          }

          cout<<"He/She is from "<<des<<",and his/her birthday is on ";
          cout<<mon<<","<<day<<","<<year<<" based on the table."<<endl;
       }

    return 0;
}
```