第一周第 1 題

Cryptanalysis is the process of breaking someone else's cryptographic writing. This sometimes involves some kind of statistical analysis of a passage of (encrypted) text. Your task is to write a program which performs a simple analysis of a given text.

Input Description：

The first line of input contains a single positive decimal integer n. This is the number of lines which follow in the input. The next n lines will contain zero or more characters (possibly including whitespace). This is the text which must be analyzed.

Output Description：

Each line of output contains a single uppercase letter, followed by a single space, then followed by a positive decimal integer. The integer indicates how many times the corresponding letter appears in the input text. Upper and lower case letters in the input are to be considered the same. No other characters must be counted. The output must be sorted in descending count order; that is, the most frequent letter is on the first output line, and the last line of output indicates the least frequent letter. If two letters have the same frequency, then the letter which comes first in the alphabet must appear first in the output. If a letter does not appear in the text, then that letter must not appear in the output.

3
This is a test.
Count me 1 2 3 4 5.
Wow!!!! Is this question easy?

S 7
T 6
I 5
E 4
O 3

Credit: 郭彥廷

```c
int main(){
    int N,i,len,j,index;
    int A[30]={0};
    char s[1024];
    scanf("%d",&N);
    fgets(s,1024,stdin);
    for(i=1;i<=N;i++){
        fgets(s,1024,stdin);
        len = strlen(s)-1;
        for(j=0; j<len; j++){
            if(!isalpha(s[j])) continue;
            s[j] = toupper(s[j]);
            index = s[j] - 'A'+1;

            A[index]++;
            //printf("%c %d\n",s[j],index);
        }
    }
    /*
    for(i=1;i<=24;i++){
        printf("%d ",A[i]);
    }
    puts("");*/
    int top;
    A[0]=0;
    char c;
    while(1){
        top=0;
        for(i=1;i<=26;i++){
            if(A[top] >= A[i]) continue;
            top = i;
        }
        if(0 == top) break;
        c = top-1+'A';
        printf("%c %d\n", c, A[top]);
        A[top]=0;
    }
    return 0;
}
```

第一周第 2 題

For a positive integer n, let f(n) denote the sum of the digits of n when represented in base 10. It is easy to see that the sequence of numbers n, f(n), f(f(n)), f(f(f(n))), . . .eventually becomes a single digit number that repeats forever. Let this single digit be denoted g(n).

For example, consider n = 1234567892.

Then:

f(n) = 1+2+3+4+5+6+7+8+9+2 = 47

f(f(n)) = 4 + 7= 11

f(f(f(n)))= 1 + 1 = 2 Therefore, g(1234567892) = 2.

Input Description:
Each line of input contains a single positive integer n at most 2,000,000,000. Input is terminated by n = 0 which should not be processed.

Output Description:
For each such integer, you are to output a single line containing g(n).

2
11
47
1234567892
0

2
2
2
2

Credit: 吳克廷

```c
int main(){
    int n;

    while(scanf("%d", &n)){
        if(n == 0) break;

        while(n > 10){
            int sum = 0;

            while(n > 0){
                sum += n % 10;
                n /= 10;
            }
            n = sum;
        }

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

第一周第 3 題

Given a range [a, b], you are to find the summation of all the odd integers in this range.

For example:

The summation of all the odd integers in the range [3, 9] is 3 + 5 + 7 + 9 = 24.

Input Description:

There can be at multiple test cases.

The first line of input gives you the number of test cases, T ($1 \leq T \leq 100$).

Then T test cases follow.

Each test case consists of 2 integers a and b ($0 \leq a \leq b \leq 100$) in two separate lines.

Output Description:
For each test case you are to print one line of output – the serial number of the test case followed by the summation of the odd integers in the range [a, b].

2
1 5
3 5

Case 1: 9
Case 2: 8

Credit: 郭彥廷

```c
int main(){
    int N,i;
    int a,b;
    scanf("%d",&N);
    for(i=1;i<=N;i++){
        scanf("%d%d",&a,&b);
        printf("Case %d: %d\n",i, (a+b)/2 * ((b-a)/2+1));
    }
    return 0;
}
```

第二周第 1 題

Superbowl Sunday is nearly here. In order to pass the time waiting for the half-time commercials and wardrobe malfunctions, the local hackers have organized a betting pool on the game. Members place their bets on the sum of the two final scores, or on the absolute difference between the two scores. Given the winning numbers for each type of bet, can you deduce the final scores?

Input Description:
The first line of input contains n, the number of test cases. n lines follow, each representing a test case.

Each test case gives s and d, non-negative integers representing the sum and (absolute) difference between the two final scores.

Output Description:
For each test case, output a line giving the two final scores, largest first.

If there are no such scores, output a line containing “impossible”.

Recall that football scores are always non-negative integers.

2
40 20
20 40

30 10
Impossible

Credit: 郭彥廷

```c
#include<stdio.h>

int main (){
    int N,i,a,b;

    scanf("%d",&N);

    for(i=1;i<=N;i++){
        scanf("%d%d", &a, &b);
        if(a<=b || (a+b)%2 !=0 || b==0){
            puts("Impossible");
            continue;
        }

        printf("%d %d\n", (a+b)/2, (a-b)/2);
    }

    return 0;
}
```

第二周第 2 題

Problems in Computer Science are often classified as belonging to a certain class of problems (e.g., NP,Unsolvable, Recursive). In this problem you will be analyzing a property of analgorithm whose classification is not known for all possible inputs.

Consider the following algorithm:

1. input n

2. print n

3. if n = 1 then STOP

4. if n is odd then n = 3n + 1

5. else n = n/2

6. GOTO 2

Given the input n 22, the following sequence of numbers will beprinted

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

It is conjectured that the algorithm above will terminate (when a 1is printed) for any integral input value. Despite the simplicity of the algorithm, it is unknown whether this conjecture is true. It has been verified, however, for all integers n such that $0 < n < 1,000,000$ (and, in fact, for many more numbers than this.) Given an input n, it is possible to determine the number of numbers printed before and including the 1 is printed. For a given n this is called the cycle-length of n. In the example above, the cycle length of 22 is 16. For any two numbers i and j you are to determine the maximum cycle length over all numbers between and including both i and j.

1 10
100 200
201 210
900 1000
0 0

1 10 20
100 200 125
201 210 89
900 1000 174

Credit: 吳克廷

```c
int main(){
    int x, y;

    while(scanf("%d %d", &x, &y)){
        if(x==0 && y==0) break;

        int max_seq = 0;
        for(int i=x; i<=y; i++){
            int n = i;
            int cnt = 1;
            while(n != 1){
                if(n%2) n = 3*n+1;
                else n /= 2;
                cnt++;
            }
            if(max_seq < cnt) max_seq = cnt;
        }

        printf("%d %d %d\n", x, y, max_seq);
    }
}
```

第二周 第 3 題

A square number is an integer number whose square root is also an integer.

For example 1, 4, 81 are some square numbers.

Given two numbers a and b you will have to find out how many square numbers are there between a and b (inclusive).

1 4
1 10
0 0

2
3

Credit: 郭彥廷

```c
int main(){
    int a,b;
    int i,total;
    while(scanf("%d%d", &a, &b) && a != 0 && b != 0){
        if(a>b) continue;
        i=1; total=0;
        while(i*i<a)    i++;

        for(i;i*i<=b;i++)
            total++;
        printf("%d\n",total);
    }
}
```

第三周第 1 題

In Act I, Leporello is telling Donna Elvira about his master's long list of conquests: "This is the list of the beauties my master has loved, a list I've made out myself: take a look, read it with me. In Italy six hundred and forty, in Germany two hundred and thirty-one, a hundred in France, ninety-one in Turkey; but in Spain already a thousand and three! Among them are country girls, waiting-maids, city beauties; there are countesses, baronesses, marchionesses, princesses: women of every rank, of every size, of every age." (Madamina, il catalogo questo) As Leporello records all the "beauties" Don Giovanni "loved" in chronological order, it is very troublesome for him to present his master's conquest to others because he needs to count the number of "beauties" by their nationality each time. You are to help Leporello to count.

Input

The input consists of at most 2000 lines. The first line contains a number n, indicating that there will be n more lines. Each following line, with at most 75 characters(if the number of characters for a line, you have to show error, and the user have to input again.), contains a country (the first word) and the name of a woman (the rest of the words in the line) Giovanni loved. You may assume that the name of all countries consist of only one word.

Output

The output consists of lines in alphabetical order. Each line starts with the name of a country, followed by the total number of women Giovanni loved in that country, separated by a space.

Sample Input

```
3
Spain Donna Elvira
England Jane Doe
Spain Donna Anna
```

Sample Output

```
England 1
Spain 2
```

Credit: 吳克廷

```cpp
#include <iostream>
#include <vector>
#include <string>
#include <cstring>
#include <cmath>
#include <iomanip>
#include <sstream>
#include <algorithm>
#include <queue>
#include <stack>
#include <set>
#include <map>
using namespace std;

int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0);

    int n;
    cin >> n;
    map <string, int> mp;
    string s;
    getline(cin, s);
    while (n--) {
        getline(cin, s);
        if(s.length() > 75){
                        n++;
                }
                else{
                        stringstream ss(s);
                ss >> s;
                mp[s]++;
                }
    }
    for (auto i: mp) {
        cout << i.first << " " << i.second << "\n";
    }
}
```

第三周第 2 題

The city of HaluaRuti has a strange hotel with infinite rooms. The groups that come to this hotel follow the following rules:

a) At the same time only members of one group can rent the hotel.

b) Each group comes in the morning of the check-in day and leaves the hotel in the evening of the check-out day.

c) Another group comes in the very next morning after the previous group has left the hotel.

d) A very important property of the incoming group is that it has one more member than its previous group unless it is the starting group. You will be given the no of members of the starting group.

e) A group with n members stays for n days in the hotel. For example, if a group of four members comes on 1st August in the morning, it will leave the hotel on 4th August in the evening and the next group of five members will come on 5th August in the morning and stay for five days and so on. Given the initial group size you will have to find the group size staying in the hotel on a specified day.

Input

The input contains round numbers S (1 ≤ S ≤ 10000) and D (1 ≤ D < 1015) in every line. S denotes the initial size of the group and D denotes that you will have to find the group size staying in the hotel on D-th day (starting from 1). All the input and output integers will be less than 1015. A group size S means that on the first day a group of S members come to the hotel and stays for S days then comes a group of S + 1 members according to the previously described rules and so on. Input is terminated by a line containing two zeroes.

Output

For each line of input, print on a single line the size of the group staying in the hotel on the D-th day.

Sample Input

1 6

3 10

3 14

0 0

Sample Output

3

5

6

Credit: 郭彥廷

```c
int main() {
    long long s, d;
    while(scanf("%lld %lld", &s, &d) && s!=0 && d!=0) {
        long long i = s, sum = 0;
        while(1) {
            sum += i;
            if(sum >= d) {
                printf("%lld\n", i);
                break;
            }
            i++;
        }
    }
    return 0;
}
```

第三周 第 3 題

A particle has initial velocity and acceleration. If its velocity after certain time is v then what will its displacement be in twice of that time?

Input

The input will contain two integers in each line. Each line makes one set of input. These two integers denote the value of v (−100 ≤ v ≤ 100) and t (0 ≤ t ≤ 200) (t means at the time the particle gains that velocity) Input is terminated by a line containing two negative integers.

Output

For each line of input print a single integer in one line denoting the displacement in double of that time.

Sample Input

0 0
5 12
-1 -1

Sample Output

0
120

Credit: 吳克廷

```c
int main(){
    int v, t;

    while(scanf("%d %d", &v, &t)){
        if(v < 0 && t < 0) break;

        printf("%d\n", v*t*2);
    }
}
```

第四周第 1 題

The “reverse and add” method is simple: choose a number, reverse its digits and add it to the original. If the sum is not a palindrome (which means, it is not the same number from left to right and right to left), repeat this procedure.

For example:

195 Initial number

591 ++

786

687 ++

1473

3741 ++

5214

4125 ++

9339 .....Resulting palindrome

In this particular case the palindrome ‘9339’ appeared after the 4th addition. This method leads to palindromes in a few step for almost all of the integers. But there are interesting exceptions. 196 is the first number for which no palindrome has been found. It is not proven though, that there is no such a palindrome.

You must write a program that give the resulting palindrome and the number of iterations (additions) to compute the palindrome.

You might assume that all tests data on this problem:

- will have an answer ,

- will be computable with less than 1000 iterations (additions),

- will yield a palindrome that is not greater than 4,294,967,295.

3
195
265
750

4 9339
5 45254
3 6666

Credit: 吳克廷

```
long long int intReverse(long long int n){
    int _sum = 0;

    while(n > 0){
        _sum *= 10;
        _sum += n%10;
        n /= 10;
    }

    return _sum;
}

int main(){
    int n;
    long long int num;
    long long int rev_n;

    scanf("%d", &n);
        for(int ss=0; ss<n; ss++){
            scanf("%lld", &num);

            rev_n = intReverse(num);

            if(rev_n == num){
                printf("%d %lld\n", 0, num);
            }
            else{
                int cnt = 0;

                while(rev_n != num){
                    cnt++;
                    num += rev_n;
                    rev_n = intReverse(num);
                }
                printf("%d %lld\n", cnt, num);
            }
        }
}
```

第四周第 2 題

Some of you may have noticed that in certain computer games, particularly the ones based on sports, the spelling of names are mutated so that they are not an exact duplicate of the real entity. This is done to avoid hassles of taking permission from each player as well as any patent issues. In this problem, you will be given a pair of names, one of which is that of a player in real life and the second found in a game. You will have to determine if the two names are same, that is the second one is obtained by mutating the first.

Two names are considered same if they are of same length and they only vary at positions where vowels occur. That means, a name which can be obtained by replacing zero or more vowels by other vowels to obtain a new name are considered same, provided they have same length.

For example, both polo and pola are same as pele but not pelet or bele.

(NOTE: vowels include a, e, i, o, u)

5
pele
polo
pele
pola
ronaldo
ronaldino
pele
pelet
pele
bele

Yes
Yes
No
No
No

Credit: 郭彥廷

```c
int isvowels(char c){
    if(c=='a') return 1;
    if(c=='e') return 1;
    if(c=='i') return 1;
    if(c=='o') return 1;
    if(c=='u') return 1;
    return 0;
}

int main (){
    int n,i,j;
    char A[30],B[30];
    scanf("%d",&n);
    for(i=1;i<=n;i++){
        scanf("%s",A);
        scanf("%s",B);
        if(strlen(A)!=strlen(B)){
            puts("No");
            continue;
        }
        int check=1;
        for(j=0;j<strlen(A);j++){
            if(A[j]==B[j]) continue;
            if(isvowels(A[j]) && isvowels(B[j]))
                continue;
            check=0;
            break;
        }
        if(check) puts("Yes");
        else puts("No");
    }
    return 0;
}
```

第四周第 3 題

Give you a integer number N (1 ≤ n ≤ 2*10^100).

Please compute

S = 1^1 + 2^2 + 3^3 + . . . + N^N

Give the last digit of S to me.

1
2
3
0

1
5
2

Credit: 吳克廷

```
int main(){
    int n;

    while(scanf("%d", &n)){
        if(n == 0) break;

        long long int _sum = 0;
        for(int i=1; i<=n; i++){
            _sum += pow(i, i);
        }

        printf("%lld\n", _sum%10);
    }
}
```

第五周第 1 題

Everybody sit down in a circle. Ok. Listen to me carefully. “Woooooo, you scwewy wabbit!” Now, could someone tell me how many words I just said?

Input
Input to your program will consist of a series of lines, each line containing multiple words (at least one). A “word” is defined as a consecutive sequence of letters (upper and/or lower case).

Output
Your program should output a word count for each line of input. Each word count should be printed on a separate line.

Sample Input
Meep Meep!
I tot I taw a putty tat.
I did! I did! I did taw a putty tat.
Shsssssssssh ... I am hunting wabbits. Heh Heh Heh Heh ...

Sample Output
2
7
10
9

```c
int main(){
        char strarr[100];
        int flag=0;
        char c;
        int count=0;
        while((c=getchar())!=EOF){
                if(c!='\n'){

                        if(c >= 'a' && c <= 'z' || c>='A' && c<='Z'){
                                if(flag == 0) count++;
                                flag = 1;

                        }else{
                                flag = 0;
                        }
                }else{
                        printf("%d\n",count);
                        count = 0;
                }

        }
        return 0;
}
```

第五周第 2 題

A character string is said to have period k if it can be formed by concatenating one or more repetitions of another string of length k. For example, the string ”abcabcabcabc” has period 3, since it is formed by 4 repetitions of the string ”abc”. It also has periods 6 (two repetitions of ”abcabc”) and 12 (one repetition of ”abcabcabcabc”). Write a program to read a character string and determine its smallest period.

Input
The first line oif the input file will contain a single integer N indicating how many test case that your program will test followed by a blank line. Each test case will contain a single character string of up to 80 non-blank characters. Two consecutive input will separated by a blank line.

Output
An integer denoting the smallest period of the input string for each input. Two consecutive output are separated by a blank line.

Sample Input
1
HoHoHo
Sample Output
2

Credit: 郭彦廷

```
int main(){
    char s[1024];
    int N,i,j,k,len;
    scanf("%d",&N);
    for(i=1;i<=N;i++){
        scanf("%s",s);
        len = strlen(s);
        while(0==len) scanf("%s",s);
        for(j=1;j<len;j++){
            for(k=0;j+k<len;k++){
                if(s[k]!=s[j+k]) break;
            }
            if(j+k==len) break;
        }
        if(len%j !=0) printf("%d",len);
        else printf("%d",j);
        puts("\n");
    }

}
```

第五周第 3 題

Please define a struct named student. In addition, the members of the student also include three variables name, scope, and fscope.

name	score
MarryHu	75
TommyChen	68
TomasChu	80

Write a function calculate that takes a struct variable as its argument and that computes the root of each person's score and then multiplies the results by 10 (stored in fscope). Finally, the main program outputs the names, the original scores, and fscores(.2f).

MarryHu 75
TommyChen 68
TomasChu 80

MarryHu 75 86.60
TommyChen 68 82.46
TomasChu 80 89.44

第六周第 1 題

Please define a structure as student contains members name (10 characters) and scope (an integer).

name	score
Marry Hu	75
Tommy Chen	68
Tomas Chu	80

Write a function compare to the data in an increasing order. Finally, output the results in the main function.

Sample output:

Tommy Chen: 68
Marry Hu: 75
Tomas Chu: 80

第六周第 2 題

Daniela is a nurse in a large hospital, which causes her working shifts to constantly change. To make it worse, she has deep sleep, and a difficult time to wake up using alarm clocks. Recently she got a digital clock as a gift, with several different options of alarm sounds, and she has hope that it might help solve her problem. But, lately, she's been very tired and want to enjoy every single moment of rest. So she carries her new clock to every place she goes, and whenever she has some spare time, she tries to sleep, setting her alarm clock to the time when she needs to wake up. But, with so much anxiety to sleep, she ends up with some difficulty to fall asleep and enjoy some rest. A problem that has been tormenting her is to know how many minutes of sleep she would have if she felt asleep immediately and woken up when the alarm clock ringed. But she is not very good with numbers, and asked you for help to write a program that, given the current time and the alarm time, find out the number of minutes she could sleep.

Input
The input contains several test cases. Each test case is described in one line, containing four integers H1, M1, H2 and M2, with H1 : M1 representing the current hour and minute, and H2 : M2 representing the time (hour and minute) when the alarm clock is set to ring ($0 \leq H1 \leq 23$, $0 \leq M1 \leq 59$, $0 \leq H2 \leq 23$, $0 \leq M2 \leq 59$). The end of the input is indicated by a line containing only four zeros, separated by blank spaces.

Output
For each test case, your program must print one line, containing a single integer, indicating the number of minutes Daniela has to sleep.

Sample Input
1 5 3 5
23 59 0 34
21 33 21 10
0 0 0 0

Sample Output
120
35
1417

Credit: 郭彥廷

```c
#include<stdio.h>

int main(){
    unsigned int H1,H2,M1,M2;
    int M;
    while(scanf("%d%d%d%d", &H1, &M1, &H2, &M2) && H1+H2+M1+M2!=0){

        M=H2*60+M2-H1*60-M1;

        if(M<0) M+=24*60;
        printf("%d\n",M);
    }
}
```

第六周第 3 題

Encoding is the process of transforming information from one format into another. There exist several different types of encoding scheme. In this problem we will talk about a very simple encoding technique; Run-Length Encoding. Run-length encoding is a very simple and easy form of data compression in which consecutive occurrences of the same characters are replaced by a single character followed by its frequency. As an example, the string 'AABBBBDAA' would be encoded to 'A2B4D1A2', quotes for clarity. In this problem, we are interested in decoding strings that were encoded using the above procedure.

Input

The first line of input is an integer T (T < 50) that indicates the number of test cases. Each case is a line consisting of an encoded string. The string will contain only digits [0-9] and letters [A-Z]. Every inputted string will be valid. That is, every letter will be followed by 1 or more digits.

Output

For each case, output the case number followed by the decoded string. Adhere to the sample for exact format. You may assume the decoded string wont have a length greater than 200 and it will only consist of upper case alphabets.

Sample Input

3
A2B4D1A2
A12
A1B1C1D1

Sample Output

Case 1: AABBBBDAA
Case 2: AAAAAAAAAAAA
Case 3: ABCD

```c
int main(){
        int n;
        scanf("%d\n",&n);
        while(n--){
        int num;
        char c;
        while(c = getchar()){
            if(c == '\n'){
                printf("\n");
                break;
            }
                scanf("%d",&num);
                while(num--){
                printf("%c",c);
            }

        }

    }
  return 0;
}
```

第六周第 4 題

Divisibility Consider an arbitrary sequence of integers. One can place + or - operators between integers in the sequence, thus deriving different arithmetical expressions that evaluate to different values. Let us, for example, take the sequence: 17, 5, -21, 15. There are eight possible expressions:

17 + 5 + -21 + 15 = 16

17 + 5 + -21 - 15 = -14

17 + 5 - -21 + 15 = 58

17 + 5 - -21 - 15 = 28

17 - 5 + -21 + 15 = 6

17 - 5 + -21 - 15 = -24

17 - 5 - -21 + 15 = 48

17 - 5 - -21 - 15 = 18

We call the sequence of integers divisible by K if + or - operators can be placed between integers in the sequence in such way that resulting value is divisible by K. In the above example, the sequence is divisible by 7 (17+5+-21-15=-14) but is not divisible by 5. You are to write a program that will determine divisibility of sequence of integers.

Input

The first line of the input file contains an integer M indicating the number of cases to be analyzed. Then M couples of lines follow. For each one of this couples, the first line of the input file contains two integers, N and K ($1 \le N \le 10000$, $2 \le K \le 100$) separated by a space. The second line contains a sequence of N integers separated by spaces. Each integer is not greater than 10000 by it's absolute value.

Output

For each case in the input file, write to the output file the word 'Divisible' if given sequence of integers is divisible by K or 'Not divisible' if it's not.

Sample Input

2

4 7

17 5 -21 15

4 5

17 5 -21 15

Sample Output

Divisible

Not divisible

Credit: 郭彥廷

```c
#include<stdio.h>
#include<stdlib.h>
int check(int A[],int i, int sum, const int *N, const int *K){
    if(i==*N) return abs(sum)%(*K);

    if(check(A,i+1,sum+A[i],N,K) == 0) return 0;
    if(check(A,i+1,sum-A[i],N,K) == 0) return 0;
    return 1;
}

int main(){
    int M, N, K, A[10010];
    int i,j;
    scanf("%d", &M);
    for(i=0;i<M;i++){
        scanf("%d%d",&N,&K);
        for(j=0;j<N;j++){
            scanf("%d",&A[j]);
        }
        if(check(A,0,0,&N,&K)==0) puts("Divisible");
        else puts("Not divisible");
    }
    return 0;
}
```

第七周第 1 題

In some places is common to remember a phone number associating its digits to letters. In this way the expression “MY LOVE” means 69 5683. Of course there are some problems, because some phone numbers cannot form a word or a phrase and the digits 1 and 0 are not associated to any letter. Your task is to read an expression and find the corresponding phone number based on the table below. An expression is composed by the capital letters (A-Z), hyphens (-) and the numbers 1 and 0.

Letters Number
ABC 2
DEF 3
GHI 4
JKL 5
MNO 6
PQRS 7
TUV 8
WXYZ 9

Input
The input consists of a set of expressions. Each expression is in a line by itself and has C characters, where $1 \leq C \leq 30$. The input is terminated by end of file (EOF).

Output
For each expression you should print the corresponding phone number, the number of capital letters, and the number of hyphens.

Sample Input
1- HOME-SWEET-HOME
2- MY-MISERABLE-JOB

Sample Output
1-4663-79338-4663 13 3
69-647372253-562 14 2

```c
#include <stdio.h>
int main() {
        char s[27] = "22233344455566677778889999";
        char c;
        int snum=0,dnum=0;
        while((c = getchar())!=EOF) {

                if(c == '\n'){
                        printf(" %d %d",snum,dnum);
                        snum=0;
                        dnum=0;
                }
                if(c >= 'A' && c<='Z'){
                        snum++;
                        printf("%c",s[c-'A']);
                }
                else{
                        if(c == '-') dnum++;
                        printf("%c",c);
                }
        }
    return 0;
}
```

第七周第 2 題

The judges from the programming contests are known to be very mean and very lazy. We, judges, want less work and more Wrong Answers! So, we'd like you to help us and write an automated judge script to judge solution runs from teams all over the world. All you have to do is write a program which receives the standard solution and a team output and gives as answer one of the following messages: "Accepted", "Presentation Error" or "Wrong Answer". We define each one as:

Accepted: As we are very mean judges, we only want you to give 'Accepted' as answer if the team output matches the standard solution integrally. That is, ALL characters must match and must be in the same order. (10188 Automated Judge Script)

Presentation Error: We want you to give 'Presentation Error' if all visible characters match but there is at least one invisible character (space or new line symbol) wrong.

Wrong Answer: If the team output could not be classified as any of the two above, then you have no option but to give 'Wrong Answer' as an answer!

Input

The input will consist of an arbitrary number of input sets. Each input set begins with a positive integer n < 100, alone in a line, which describes the number of lines of the standard solution. The next n lines contain the standard solution. Then there is a positive integer m < 100, alone in a line, which describes the number of lines of the team output. The next m lines contain the team output. The input is terminated by a value of n = 0, and should not be processed. No line will have more than 120 characters.

Output

For each set you should output one of the following lines:

Run #x: Accepted y

Run #x: Presentation Error y

Run #x: Wrong Answer y

Where x stands for the number of the input set (starting from 1), and y is the total number of characters (including spaces, but not including the symbols for new lines) in the standard solution.

Sample Input

```
2
The answer is: 10
The answer is: 5
2
The answer is: 10
The answer is: 5
2
The answer is: 10
The answer is: 5
2
The answer is: 10
The answer is: 15
2
The answer is: 10
The answer is:    5
2
The answer is: 10
The answer is: 5
3
Input Set #1: YES
Input Set #2: NO
Input Set #3: NO
3
Input Set #0: YES
Input Set #1: NO
Input Set #2: NO
1
1 0 1 0
1
1010
1
The judges are mean!
1
The judges are good!
0
```

Sample Output

```
Run #1: Accepted 33
```

Run #2: Wrong Answer 33
Run #3: Presentation Error 34
Run #4: Wrong Answer 49
Run #5: Presentation Error 7
Run #6: Wrong Answer 20

Credit: 郭彥廷

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

typedef enum {AC, WA, PE}status;

void printState(status Status, int runtime, int count){
    switch (Status)
    {
    case AC:
        printf("Run #%d: Accepted %d\n", runtime, count);
        break;
    case WA:
        printf("Run #%d: Wrong Answer %d\n", runtime, count);
        break;
    case PE:
        printf("Run #%d: Presentation Error %d\n", runtime, count);
        break;
    default:
        break;
    }
}
```

```
status check(char a[], char s[]){
    status Status = AC;
    int len_s, len_a, i_a=0,i_s=0;
    len_a = strlen(a)-1;
    len_s = strlen(s)-1;

    while(i_a<len_a && i_s<len_s){
        if(a[i_a] != s[i_s] && (isspace(a[i_a]) || isspace(s[i_s]))){
            while (isspace(a[i_a]) && i_a<len_a) i_a++;
            while (isspace(s[i_s]) && i_s<len_s) i_s++;
            Status = PE;
        }
        if(a[i_a] != s[i_s]) return WA;
        i_a++; i_s++;
    }

    while(i_a<len_a){
        if(!isspace(a[i_a++])) return WA;
        Status = PE;
    }

    while(i_s<len_s){
        if(!isspace(a[i_s++])) return WA;
        Status = PE;
    }

    return Status;
}
```

```
int main(){
    int n,m;
    int i=0,j,runtime=0, count;
    char sol[2000], ans[2000], tmp[200];
    status Status;

    while(scanf("%d",&n) && n != 0){
        count = 0; Status = AC;
        memset(sol,'\0',sizeof(sol));
        memset(ans,'\0',sizeof(ans));

        fgets(tmp,200,stdin);//clean buffer
        for(i=0;i<n;i++){
            fgets(tmp,200,stdin);
            strcat(sol,tmp);
            count += strlen(tmp)-1;
        }

        scanf("%d",&m);
        fgets(tmp,200,stdin);//clean buffer
        for(i=0;i<m;i++){
            fgets(tmp,200,stdin);
            strcat(ans,tmp);
        }
        Status = check(ans, sol);

        if(Status == WA){
            printState(WA, ++runtime, count);
            continue;
        }

        printState(Status, ++runtime, count);

    }
}
```

第七周第 3 題

We are familiar with the Fibonacci sequence (1, 1, 2, 3, 5, 8, ...). What if we define a similar sequence for strings? Sounds interesting? Let's see. We define the following sequence: BF S(0) = 0 BF S(1) = 1 (here "0" and "1" are strings, not simply the numerical digit, 0 or 1) for all (n > 1) BF S(n) = BF S(n − 2) + BF S(n − 1) (here, '+' denotes the string concatenation operation). (i.e. the n-th string in this sequence is a concatenation of a previous two strings). So, the first few strings of this sequence are: 0, 1, 01, 101, 01101, and so on. Your task is to find the N-th string of the sequence and print all of its characters from the i-th to j-th position, inclusive. (All of N, i, j are 0-based indices) (12041 BFS (Binary Fibonacci String))

Input

The first line of the input file contains an integer T (T ≤ 100) which denotes the total number of test cases. The description of each test case is given below: Three integers N, i, j (0 ≤ N, i, j ≤ 2^31 − 1) and (i ≤ j and j − i ≤ 10000). You can assume that, both i and j will be valid indices (i.e. 0 ≤ i, j < length of BF S(N)).

Output

For each test case, print the substring from the i-th to the j-th position of BF S(N) in a single line.

Sample Input

```
3
3 1 2
1 0 0
9 5 12
```

Sample Output

```
01
1
10101101
```

```cpp
#include <stdio.h>
#include <algorithm>
using namespace std;
long long f[50];
void fib_bit(int n, long long l, long long r) {
    if(l > r)   return;
    if(n == 0)  putchar('0');
    else if(n == 1)  putchar('1');
    else {
        if(l < f[n-2]) {
            fib_bit(n-2, l, min(r, f[n-2]-1));
            fib_bit(n-1, 0, r-f[n-2]);
        } else {
            fib_bit(n-1, l-f[n-2], r-f[n-2]);
        }
    }
}
int main() {
    int t, i;
    long long n, a, b;
    f[0] = 1, f[1] = 1;
    for(a = 2; a <= 50; a++)
        f[a] = f[a-1]+f[a-2];
    scanf("%d", &t);
    while(t--) {
        scanf("%lld %lld %lld", &n, &a, &b);
        if(n >= 48)
            n = 48 - (n%2);
        fib_bit(n, a, b);
        puts("");
    }
    return 0;
}
```

第七周第 4 題

Please define a structure as student containing name (10 characters), scope (an integer), and enum sex{Female, Male}.

name	score	sex
Marry Hu	85	Female
Tommy Chen	68	Male
Tomas Chu	80	Male
Kitty Lee	75	Female

Write a function compare to output the maximum scopes for Female and Male, respectively.

Input
name (10 characters), score (an integer), and enum sex{Female, Male} separated by Tabs

Output
name, score, and sex
separated by spaces

Tommy Chen 68 Male
Marry Hu 85 Female
Tomas Chu 80 Male
Kitty Lee 75 Female

Marry Hu 85 Female
Tomas Chu 80 Male