National Collegiate Programming Contest 1996 (Taiwan)

Problem A: Area of an Object

The Problem

Assume that there is an object in a digital picture whose boundary is specified by a sequence of moves. A move can only be in the 8 directions to its neighbours. The 8 directions are defined as follows.

```
3 2 1
4 . 0
5 6 7
```

A sequence of moves represented by the numbers 0, 1, 2, 3, 4, 5, 6, 7 are called a *chain code*. Assume that the object has no holes in it. An object in a digital picture can then be specified by a starting point, followed a chain code. For example, with starting point $(1, 5)$, the object shown at the left of the following figure can be represented by a string at the right.

```
0000X0000000X
000XXXXX0000X
XXXXXXXXXXXX0                       1,5 55440007007172101265444344 3
0000XXXXXX000
0000000X0X000
```

Write a program to calculate the area of the object in a picture by counting how many X's are covered by the object.

The Input

Each object in the picture is represented by a starting point and followed by a chain code. The chain code may be more than 80 characters.

The Output

If the chain code is a sequence of moves from the starting point back to the same point, then print out the area of the object represented by the chain code, otherwise, print the error message "Illegal chain code."

Sample Input

```
1,40 5445456667770100110700071113334432544353*
1,24 5676003124*
```

Sample Output

```
object 1: the area is 123
object 2: Illegal chain code.
```

Problem B: Binary String Combination

The Problem

Consider n-*bit* binary string with exact k 1's. Obviously, there are $\binom{n}{k}$ binary strings with this property. For example, the list of 5-bit binary strings with two 1's is as follows:

00011, 00101, 00110, 01001, 01010, 01100, 10001, 10010, 10100, 11000

There are totally $\binom{5}{2} = 10$ binary strings in the above example. The above strings are arranged in ascending order by their values. In our example, 00011 is the first string since it is the smallest, 00101 is the second string since it is bigger than 00011 and smaller than any others. Finally, 11000 is the tenth string since it is the largest.

Given n, k, and m, write a program to generate the m-th n-bit binary string with exact k 1's.

The Input

The input file may contain many instances. Each instance consists of three numbers, n, k, and m. You may assume that $n \leq 32$.

The Output

For each instance, print out the m-th n-bit binary string with exactly k 1's, and its corresponding decimal number.

Sample Input

```
7 3 10
16 8 10001
32 12 5000000
```

Sample Output

```
n = 7, k = 3, m = 10
Decimal number = 28
Binary string = 0011100

n = 16, k = 8, m = 10001
Decimal number = 49897
Binary string = 1100001011101001

n = 32, k = 12, m = 5000000
Decimal number = 31475149
Binary string = 00000001111000000100010111001101
```

Probelm C: Find the Rank

The Problem

Given two sorted sequences of integers $A = a_1, a_2, \ldots, a_m$ and $B = b_1, b_2, b_n$. Both sequences are sorted in nondecreasing order. Let $s = \left\lceil \frac{m}{2} \right\rceil$ and $t = \left\lceil \frac{n}{2} \right\rceil$. Obviously, $x = a_s + b_t$ is one of the nm elements $c_{i,j} = a_i + b_j$, where $1 \leq i \leq m, 1 \leq j \leq n$. Write a program to find the rank of x when all these nm elements are sorted in nondecreasing order. In case that there are more than one element which have the same value as x, let the rank of x be the first of these elements.

The Input

The input file may contain more than one instance. Each instance consists of two sorted lists. The first number in each list is the number of elements in that list. This is then followed by the elements of that lists, sorted in nondecreasing order. You may assume that the maximum number of elements in each list is no more than 2000.

Output

For each instance, your program must print the numbers m and n, together with the rank of $a_s + b_t$ in a format as shown in the Sample Output.

Sample Input

```
5
15 20 34 36 71
4
18 25 50 100
8
100 101 105 151 151 180 200 240
5
80 88 100 200 202
```

Sample Output

```
m=5, n=4, rank of 59 is 7
m=8, n=5, rank of 251 is 14
```

Problem D: Minimum Connection Cost

The Problem

Let V be a set of points $v_1, v_2, \ldots, v_n$ in the plane, where $v_i = (x_i, y_i)$, $1 \leq i \leq n$. Assume that all the x_i's and the y_i's are integers less than or equal to 255. Two points v_i and v_j are *connected* if either there is a line between them, or there is another point v_k such that both the pair of points v_i and v_k and the pair of points v_k and v_j are connected. Given a set of points, we want to construct a set of lines to make every pair of points connected. It is easy to see that $n-1$ lines are necessary and sufficient to connect n points. Assume that the cost of the line connecting any pair of points v_i and v_j is $(x_i - x_j)^2 + (y_i - y_j)^2$.

For example, let the set of points be $\{(100, 100), (20, 10), (20, 20), (10, 20)\}$. The four lines $(100, 100)$ to $(20, 20)$, $(20, 10)$ to $(20, 20)$, and $(20, 20)$ to $(10, 20)$ connects all pair of the given points. The cost of the connection is $12800 + 100 + 100 = 13000$.

Write a program to compute the cost of connecting a given set of points.

The Input

The input file contain more than one instance. Each instance is a set of points in the plan. The first number of an instance is n, the number of points in the plan to be connected. This is then followed by n points. Each point consists of two numbers, x and y.

The Output

For each instance, print the minimum cost to connect the set of points. Note that you need not print the lines that connect these points.

Sample Input

```
2
0 1
1 0
4
100 100
20 10
20 20
10 20
```

Sample Output

```
The minimum connection cost is 2.
The minimum connection cost is 13000.
```

Problem E: Number of Combinations

The Problem

The number of combinations of n distinct items taken m at a time is given by the following formula:

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}$$

for integers m and n, with $0 \leq m \leq n$. Given two numbers n and m, write a program to compute the value of $\binom{n}{m}$.

The Input

The input file may contain more than one instance. Each instance consists of two integers m and n with $0 < m < n$.

The Output

The values of n, m, and $\binom{n}{m}$ must be printed in a format as shown in the Sample Output. Note that if the number of digits of $\binom{n}{m}$ is more than 50, it should be warped to the next line.

Sample Input

```
5 2
3 3
123 4
```

Sample Output

```
C(5,2)=10
C(3,3)=1
C(123,4)=9078630
```

Problem F: Euler Tour

The Problem

Let G be a simple undirected graph with n vertices $1, 2, \ldots, n$. The graph G can be represented by an *adjacency matrix* A. Each element, $a_{i,j}$, of A is either 0 or 1, and $(a_{i,j}) = 1$ if, and only if, there is an edge between vertex i and vertex j. For example, the following picture shows a graph at the left and its adjacency matrix at the right.

```
1----2
 \  /                     0 1 1 0 0
  \/                      1 0 1 0 0
  3                       1 1 0 1 1
  /\                      0 0 1 0 1
 /  \                     0 0 1 1 0
4----5
```

A tour of G is a sequence $v_0, e_1, v_1, e_2, v_2 \ldots, e_k, v_k$, Such that e_i is an edge that connects v_{i-1} and v_i. An Euler tour is a tour $v_0, e_1, v_1, e_2, v_2 \ldots, e_k, v_k$ such that $v_0 = v_k$, and each edge appears exactly once in the tour. Since the graph is simple, we will omit all the edges in the tour. For example, the sequence 2,3,5,4,3,1,2 is an Euler tour of the above graph. In this problem, we want to write a program to construct an Euler tour for simple undirected graphs. Note that not every graph has an Euler tour, and if a graph has an Euler tour, the tour may not be unique.

The Input

The input file may contain more than one instance. The first line of each instance is the number of vertices n. It is then followed by $a_{i,j}$'s, arranged in the order $a_{1,j}$, $j = 1, 2 \ldots, n$, $a_{2,j}$, $j = 1, 2 \ldots, n$, and so on. You may assume that $n < 100$.

The Output

If the graph has an Euler tour, the output is a sequence of vertices which is an Euler tour of the graph. If the graph has no Euler tour, print the string "The graph is not Eulerian."

Sample Input

```
5
0 1 1 0 0
1 0 1 0 0
1 1 0 1 1
0 0 1 0 1
0 0 1 1 0
3
0 1 0
1 0 1
0 1 0
```

Sample Output

```
2 3 5 4 3 1 2
The Graph is not Eulerian.
```