Problem A

Special Sequence Generation

The Problem

A sequence a_1, a_2, ..., a_n is called a special sequence of order if it satisfies the following conditions.

1. $a_i \le a_{i+1}$ $(1 \le i < n)$, and
2. let $x_k = \sum_{i=1}^{k} a_i$, then $x_k \ge k^2$ for $k = 1, 2, ..., n-1$, and $x_n = \sum_{i=1}^{n} a_i = n^2$

For example. 1,3,5 and 2,2,5 are special sequences of order 3. However, 1,2,6 is not a special sequence of order 3 because the sum of the first two elements is less than 2^2. In this problem, you are going to write a program to generate special sequences.

The Input

Each line of the input file contains a positive number n, which is the order of the special sequence to be generated. You may assume that $n < 20$.

The Output

For each input number n, generate all special sequences of n. Each special sequence must be printed in a line. The sequence generated must be in lexicographic order. That is a_1, a_2, ..., a_n appears before a'_1, a'_2, ..., a'_n, if $q < n$, for some $x_k = x'_k$, for $k = 1, 2, ..., q-1$, and $x_q < x'_q$. Your program is not supposed to print all sequences if the number of sequences is greater then 40. Your program must print the first 20 sequences, followed by "`...`", and then followed by the last 20 sequences only. In addition to the sequences, your program must print the total number of sequences of order at the last line.

Sample Input

```
3
5
0
```

Sample Output

```
1 3 5
1 4 4
2 2 5
2 3 4
3 3 3
total 5 sequences of order 3
1 3 5 7 9
```

```
1 3 5 8 8
1 3 6 6 9
1 3 6 7 8
1 3 7 7 7
1 4 4 7 9
1 4 4 8 8
1 4 5 6 9
1 4 5 7 8
1 4 6 6 8
1 4 6 7 7
1 5 5 5 9
1 5 5 6 8
1 5 5 7 7
1 5 6 6 7
1 6 6 6 6
2 2 5 7 9
2 2 5 8 8
2 2 6 6 9
2 2 6 7 8
...
3 3 4 7 8
3 3 5 5 9
3 3 5 6 8
3 3 5 7 7
3 3 6 6 7
3 4 4 5 9
3 4 4 6 8
3 4 4 7 7
3 4 5 5 8
3 4 5 6 7
3 4 6 6 6
3 5 5 5 7
3 5 5 6 6
4 4 4 4 9
4 4 4 5 8
4 4 4 6 7
4 4 5 5 7
4 4 5 6 6
4 5 5 5 6
5 5 5 5 5
total 59 sequences of order 5
```

Problem B

Activity Selection Problem

The Problem

Suppose we have a set of proposed activities, {1, 2, ..., n}, that wish to use the same hall. Each activity has a start time s and a finish time f, where $s < f$. If activity is selected, it takes place during the time interval [s, f). Assume that the hall can be used by only one activity at a time. Write a program to select maximum number of activities that can be scheduled in the hall.

The Input

Each set of activities begins with a positive number n, which is the number of activities. It is then followed by pairs of positive numbers. Each pair of numbers and are the start time s and the finish time f of activity . You may assume that all numbers are integers and that $n < 100$.

The Output

For each set of activities, print the maximum number of activities that can be scheduled in one hall, followed by the list of the activities selected by your program. Print the list of selected activities in increasing order, and print at most 30 activities in a line.

Sample Input

```
3
1 3
2 4
3 6
5
8 9
6 10
3 8
9 11
5 7
0
```

Sample Output

```
At most 2 activities can be scheduled.
1 3
At most 3 activities can be scheduled.
1 4 5
```

Problem C
The Shortest Path Problem

The Problem

Given a weighted, directed graph $G = (V, E)$ with weight functions $w(v_i, v_j)$: $w(v_i, v_j) = A_{i,j}$, a source vertex v_1, and a sink vertex v_n. A path from v_1 to v_n is a sequence of vertices $p = v_1, v_x, ..., v_n = e(v_1, v_x), e(v_x, v_y), ..., e(v_z, v_n)$, such that $e(v_i, v_j)$ is an edge of v_i, v_j for $i, j = 1, 2, ..., n$. We assume that $w(v_i, v_j) > 0$ for every $i \neq j$. The weight of a path $w(p) = w(v_1, v_x, ..., v_n)$ is the sum of the weights of its edges:

$$w(v_1, v_x, ..., v_n) = w(v_1, v_x) + w(v_x, v_y) + ... + w(v_z, v_n)$$

The shortest path p from v_1 to v_n is a path such that $w(p) \leq w(r)$ for every path from v_1 to v_n. The second shortest path q from v_1 to v_n is a path such that for any path from v_1 to v_n, other than p, $w(q) \leq w(r)$. Note that the second shortest path q from v_1 to v_n might be a shortest path from v_1 to v_n if the shortest path from v_1 to v_n is not unique. Although the shortest path p and the second shortest path q may share some edges, they must not be the same path.

The Input

The first line of each instance is n the number of vertices in G. It is then followed by the values of $w(v_i, v_j)$. These $w(v_i, v_j)$ appear in row major order: (1, 1), (1, 2), ..., (n, 1), (2, 1), (2, 2), ..., (n, n). If $e(v_i, v_j)$ is not an edge, then $w(i, j) = 0$. You may assume that the number of vertices is at most 100.

The Output

Assume that the vertex of are labeled with {1,2, ..., n}. The source vertex v_1 is 1 and the sink vertex v_n is n. For each instance, print the weight of the shortest path and the weight of the second shortest path from v_1 to v_n.

Sample Input

```
3
0 1 3
1 0 1
3 1 0
4
0 3 1 0
0 0 2 2
0 1 0 4
0 0 0 0
0
```

Sample Output

```
case 1: min=2, 2nd=3
case 2: min=4, 2nd=5
```

Problem D
Reflected Gray Codes

The Problem

Consider the n-bit reflected Gray code x, $x = \overline{a_1 a_2 \ldots a_n}$ x can be generated by with the following formula:

$a_1 = x \div 2^{n-1}$,

$a_i = ((x + 2^{n-i}) \bmod 2^{n-i+2}) \div 2^{n-i+1}$, $i > 1$

where is the reverse of all the code words of . There are in total 2^n n-bit binary strings for x. The first string represents decimal 0, the second string represents decimal 1, ..., and the last one represents decimal $2^n - 1$. For example, the list is shown as follows:

0000 (0)
0001 (1)
0011 (2)
0010 (3)
0110 (4)
0111 (5)
0101 (6)
0100 (7)
1100 (8)
1101 (9)
1111 (10)
1110 (11)
1010 (12)
1011 (13)
1001 (14)
1000 (15)

The number in the ( ) is the corresponding decimal for the Gray code.

We are interested in the n-bit reflected Gray codes with exact m 1's. Obviously, there are in total $C(n, m)$ binary strings satisfy the requirement. For example, the list of 4-bit reflected Gray codes with two 1's is as follows:

0011, 0110, 0101, 1100, 1010, 1001

There are totally $C(4, 2)$=6 binary strings in the above example. Those strings are ranked in ascending order according to the corresponding decimal number of that reflected Gray code. In other words, in our example, 0011 (2) is called the first string, 0110 (4) is the second string, and so on. Finally, 1001 (14) is the sixth string. Given n, m, and p, write a program to generate the p-th n-bit reflected Gray code with exact m 1's.

The Input

Each instance contains three integers, n, m, and p. You may assume that $n \leq 32$, and $p \leq \min\{2^n - 1, C(n, m)\}$.

The Output

For each instance, print out the reflected Gray code and its corresponding decimal number.

Sample Input

```
4 2 4
5 3 6
6 2 3
0 0 0
```

Sample Output

```
Decimal number = 8
reflected Gray code = 1100
Decimal number = 19
reflected Gray code = 11010
Decimal number = 6
reflected Gray code = 000101
```

Problem E

Compute the Value of π

The Problem

The following formula is an approximation of π:

(Lost formula)

On input two positive integers n and m, write a program to compute an m-digit approximate value of π by the above formula.

The Input

Each instance contains two positive integers, n and m. You may assume that both n and m are less than 200.

The Output

For each instance n and m, compute (Lost formula) up to m decimal digits after the decimal point. Print at most 80 digits in a line.

Sample Input

```
50 8
100 50
0 0
```

Sample Output

```
3.14159264
3.14159265358979323846264338327950288419716939937504
```

Problem F

Hill Cipher

Problem G

Permutation Enumeration

Problem H

Evaluate a Boolean Function