Princes

Having the personal treasury fully filled, Sultan of Istanbul decided to share out his diamonds among his three princes nicknamed A, B and C. The sharing out must be done in a ceremony following certain rules. To this end, after having the three princes sat down in front of him, Sultan started to hand in **N** diamonds (numbered 1,2,…,**N**) one-at-a-time manner in **N** steps. The value of *i* th diamond is 2^{i-1}, hence the value range of diamonds are 2^0, 2^1, 2^2,..., 2^{N-1}.

Princes have no diamonds at the beginning. Let a_j, b_j and c_j be the respective total value of diamonds given to princes A, B, and C right after step j=1,2,...,**N**. Since prince A is the eldest and C is the youngest, the following inequality must be satisfied at the end of every step of the sharing out.

$$a_j \geq b_j \geq c_j \quad \text{for } j=1,2,...,\mathbf{N}$$

Any sharing out satisfying the above inequality is called *fair*. Let's use $\mathtt{d}_j$ to indicate the diamond number and $\mathtt{p}_j$ to indicate prince nickname involved at step *j*. Then fair sharing outs are encoded with the vector "$\mathtt{p}_1\mathtt{d}_1\ \mathtt{p}_2\mathtt{d}_2\ \ldots\ \mathtt{p}_N\mathtt{d}_N$". As there are too many distinct fair sharing outs, Sultan is interested in **K***th* of them. When the codes of all fair sharing outs are sorted in ascending order, **K***th* fair sharing out is the one having order **K** in this list. The criteria for ordering fair sharing outs are provided through the example below.

Given **N** and **K**, your task is to find the code of the **K***th* fair sharing out.

Example

Let **N**=4 and suppose Sultan does the following sharing out.

Step (*j*=1): Hand in diamond 3 to prince A
Step (*j*=2): Hand in diamond 2 to prince B
Step (*j*=3): Hand in diamond 4 to prince A
Step (*j*=4): Hand in diamond 1 to prince C

The above sharing out is fair since at the end of every step above inequality is satisfied. The code for this fair sharing out is "`A3 B2 A4 C1`".

The sharing out ordering criteria is first based on increasing alphanumeric order of prince nicknames (A<B<C), and then increasing order of diamond numbers (1<2<…<N). For instance the distribution with the code "`A3 A4 A1 B2`" precedes the distribution code "`A3 B2 A4 C1`". Note that code of the **1***st* fair distribution is "`A1 A2 A3 A4`", while the **2***nd* is "`A1 A2 A4 A3`". Your task is to find the **K***th* of them.

For **N**≥10, for instance, the code "… `A8` …" precedes the code "… `A12` …" as number comparisons (not string comparison) are done on diamond numbers (i.e. 8 is smaller than 12).

Task

Please write a program that finds the code of **K***th* fair sharing out. The input/output format is explained below with three sample input/outputs.

input file (`Standard Input`)	output file (`Standard Output`)
`3 1`	`A1 A2 A3`
`3 3`	`A1 A3 B2`
`4 9`	`A1 A4 A2 B3`

Input format: (Only one line) The line has two space-separated integers (**N** and **K**, in this order).

Output format: (Only one line) The code of the **K***th* fair sharing out.

*On the second sample above, 1st fair sharing out is "`A1 A2 A3`" and the 2nd fair sharing out is "`A1 A3 A2`", and the 3rd one is "`A1 A3 B2`".

Subtasks

Subtask 1 (11 points)

1≤**N**≤20

1≤**K**≤10^{18}

Subtask 2 (17 points)

1≤**N**≤50
1≤**K**≤10^{200}

Subtask 3 (31 points)

1≤**N**≤100
1≤**K**≤10^{19}

Subtask 4 (41 points)

1≤**N**≤100
1≤**K**≤10^{300}

Implementation details

You have to submit only one file, called `princes.c`, `princes.cpp` or `princes.pas`. The file implements your full program.