



Array of **N** integers is given. In one move Mirko can decrease one of its elements by one, and repeat it arbitrarily but no number can be less than 1. He is given **T** moves and has to use them all. He wants that after he is done maximal absolute difference of any two neighbouring numbers is minimal.

INPUT DATA

In first line of standard input there are two integers **N** and **T** ($2 \leq N \leq 100\,000$, $1 \leq T \leq 10^9$), array length and number of moves Mirko should make.

In next line there are **N** natural numbers less than 10^9 .

OUTPUT DATA

In first and only line output array after Mirko has done **T** moves.

SCORING

For every test case your solution scores points as follows: Let **D** be maximal absolute difference of any two neighbouring numbers in your solution and **S** in official(optimal) solution, then your score for that test case is $(10 * (S + 1) / (D+1))$.

So for example, if your solution has difference **D** equal to optimal one, your score is 10 points. If your solution output array which cannot be made in **T** moves or there was some other error during the program execution your score is 0 on that test case.

TEST EXAMPLES

input 5 2 3 2 3 2 2	input 5 5 4 2 3 7 6	input 3 6 10 10 1
output 2 2 2 2 2	output 3 2 3 4 5	output 9 5 1

In second test example, for printed solution $\{ 4\ 2\ 3\ 5\ 6 \}$ maximal difference of any two neighbouring numbers is 2, so score is $10 * (2/3) = 6.66$ points, because optimal solution is 1.