Problem 1: Papyrus

Jones is pursuing his life-long passion of exploring ancient ruins. Currently, he is investigating the writings of an ancient Amazon civilization. Deep in a mysterious pyramid he has discovered a papyrus with a rather long inscription. Jones believes that, deciphering this papyrus will lead him to a great treasure or at least a way out of the pyramid.

Thus far, by comparing with others texts, he has discovered that:

- The ancients used exactly 9 hieroglyphs depicting various objects/animals (e.g. an eagle, a stone, the sun, etc.). Jones has associated each of these symbols with a number (from 1 to 9).
- Mysteriously, there is a way to map each of the symbols to a character in the English alphabet (only uppercase letters). For example, ‘A’ corresponds to symbol ‘1’, ‘B’ to symbol ‘2’, ‘C’ to ‘3’, …, ‘Z’ to symbol ‘26’.
- The ancients had no love for whitespaces. For example, the sequence ‘26’ may be interpreted as ‘Z’ but it can also mean ‘AF’

Jones wants to compute in **how many ways** he can translate the sequence of hieroglyphs he found on the papyrus. Since this value can be very large, print this number **modulo (10^7 + 1)**.

Data Format

Input

The input file is called “*papyrus.in*”.

The input file will contain only one line with the sequence of numbers between 1 and 9, each representing a special hieroglyph.

Output

The output file is called “*papyrus.out*”.
It contains only one line, which is the number of ways in which Jones can translate the string, **modulo (10^7 + 1).**

Data Limits

1 <= length(input_string) <= **100.000**
For 40% of the tests, length(input_string) <= 20.

Time limits: C++ (0.4s), Java (0.8s)

Note

When submiting your source file, please name it **papyrus.cpp/Papyrus.java.**

Example 1

papyrus.in	papyrus.out
226	3

Explanation

226 can be translated as: BBF, BZ or VF.

Example 2

papyrus.in	papyrus.out
10205	1

Explanation

10205 can only be translated as JTE.

Problem 2: Interview

Jones is preparing everyday for programming interviews. He has acquired a very popular book with interview problems and is now trying to solve all the **N** assignments in the book **in the order that they appear**. For each assignment, he knows exactly how much time he will need to solve it.

Note: Jones doesn't want to work on the same problem for more than one day, therefore he won't stop working on a problem until he finishes it.

Jones is optimistic he will be able to finish all **N** assignments before the next interview, which will be in **K** days from today. Help Jones estimate the **minimum** amount of time he needs to spend every day solving problems to succeed.

Data Format

Input

The input file is called "*interview.in*".

The first line contains the numbers **N** and **K**.
The second line contains **N** integers denoting the amount of time $\mathbf{t_i}$ needed to solve each problem.

Output

The output file is called "*interview.out*".
It contains only one line with one integer **T**, which is the minimum amount of time Jones needs to spend every day to solve all **N** assignments in **K** days.

Data Limits

$1 <= K <= N <= 100.000$
$1 <= t_i <= 10^{12}$

For 40% of the tests: $1 <= N <= 1000$; $1 <= K <= 100$; $1 <= t_i <= 100$

Time limits: C++ (0.2s), Java (0.5s)

Note

When submitting your source file, please name it ***interview.cpp*/*Interview.java*.**

Hint: Try to guess the correct answer. (e.g. Guess that the correct answer is T then check if you can finish all tasks if you spend at most T time units everyday solving problems)

Hint: Since the answer can be a very large number, you need to represent it using a data type that supports 64 bits (C++ - long long, Java - long)

Example

interview.in	**interview.out**
5 3 1 2 2 1 2	3

interview.in	interview.out
10 5 3 4 2 1 6 7 1 2 2 3	8

Explanation - Example 1

We have to solve 5 tasks in 3 days.

Solution:
Day 1: Solve task 1 and 2. ---- 1 + 2 = 3
Day 2: Solve task 3 and 4. ---- 2 + 1 = 3
Day 3: Solve the remaining task, task 5. ---- 2

The minimum amount of time spent every day is 3.
Jones cannot spend less time **each** day or he will never finish all the assignments in time.

Explanation - Example 2

We have to solve 10 tasks in 5 days.

Solution:
Day 1: Solve task 1 and 2. ---- 3 + 4 = 7
Day 2: Solve task 3 and 4. ---- 2 + 1 = 3
Day 3: Solve task 5. ---- 6
Day 4: Solve task 6. ---- 7
Day 5: Solve task 7, 8, 9 and 10. ---- 1 + 2 + 2 + 3 = 8

The minimum amount of time spent every day is 8.
Jones cannot spend less time **each** day or he will never finish all the assignments in time.