Logging Compression

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

A system contains n types of programs, indexed from 1 to n. Once an instance of program type k starts, it generates a log entry every k seconds—specifically, at times $k, 2k, 3k, \ldots$ relative to its start time.

At time 0, the system executes m commands simultaneously: the i-th command launches one instance for each program type with an index in the range $[l_i, r_i]$. Note that if the ranges of multiple commands overlap, multiple instances of the same program type may be launched, and each instance generates logs independently.

The system records all log entries generated at times $1, 2, \ldots, n$. Each record contains only the index of the program type corresponding to the instance that generated the log.

Your task is to design an algorithm to compress these logs: you must assign a non-empty binary string as an identifier to all the n program types, ensuring that no identifier is a prefix of another. The cost of encoding a single record is the length of the identifier assigned to the corresponding program type, and the total cost is the sum of the costs of all records. Find the minimum possible total cost.

Input

The first line contains two integers n, m ($1 \le n \le 10^{10}$, $1 \le m \le 10^5$), representing the number of types of programs and the number of commands.

Each of the next m lines contains two integers l_i, r_i ($1 \le l_i \le r_i \le n$), representing a start command.

Output

Print a single integer: the minimum total cost.

Examples

standard input	standard output
5 1 1 5	20
6 2 1 3 4 6	32

Note

Assume that logs are recorded in chronological order; for logs generated at the same time, the records follow a monotonically non-decreasing order based on the index of the corresponding program types.

For the first sample, the logs are $[1, 1, 2, 1, 3, 1, 2, 4, 1, 5]$. An optimal assignment of identifiers is (program 1: "0", program 2: "100", program 3: "101", program 4: "110", program 5: "111"). The cost

is $5 \times 1 + 2 \times 3 + 1 \times 3 + 1 \times 3 + 1 \times 3 = 20$.

For the second sample, the logs are $[1, 1, 2, 1, 3, 1, 2, 4, 1, 5, 1, 2, 3, 6]$. An optimal assignment of identifiers is (program 1: "`0`", program 2: "`10`", program 3: "`1100`", program 4: "`1101`", program 5: "`1110`", program 6: "`1111`"). The cost is $6 \times 1 + 3 \times 2 + 2 \times 4 + 1 \times 4 + 1 \times 4 + 1 \times 4 = 32$.