2018.1.11 – Adaptive WBS

For the compression of a binary string one uses the Adapted White Block Skipping method - ignoring (or skipping) the uniform blocks (group of bits) (blocks that are composed only by bits having the same value, 0 or 1). The string is split into equal blocks (group of bits) that contain **n** bits; the blocks will be coded independently. There are two possible condings: the coding in which the blocks composed by 0 bits are ignored, and the coding in which the blocks composed by bits of 1 are ignored. In the end one choses the coding that produces the maximal compression ratio. If the two compression versions produce the same compression ratio, one will use the version that ignores the blocks composed only of 0.

For the coding that ignores the blocks composed only from bits of 0, the compressed string begins with a bit of 0, followed by the codes that correspond to the blocks from the original stream, computed according to the following rule: if all the bits in the block are 0, the block is replaced by a single bit of 0; if at least one bit from the block is 1, then the bits from the block are copied and are prefixed by a bit of 1 (in other words, one bit of 1 is added at the start of that block).

For the coding that ignores the blocks composed only from bits of 1, the compressed string begins with a bit of 1, followed by the codes that correspond to the blocks from the original stream, computed according to the following rule: if all the bits in the block are 1, the block is replaced by a single bit of 1; if at least one bit from the block is 0, then the bits from the block are copied and are prefixed by a bit of 0 (in other words, one bit of 0 is added at the start of that block).

Requirement

Given a positive number **N** representing the number of elements in the string, then the positive number **n** representing the number of elements from each block to be coded, then the **N** elements of the string, one should generate the compressed (coded) string and the compression ratio (the ratio between the number of bits in the initial string - **N** - and the number of bits in the compressed string). If the input string cannot be exactly partitioned into **n** bits blocks, the last remaining bits are coded as they are forming a non-uniform block (but without adding any supplemental bits for completing the block).

Input data

On the first line is the positive integer number **N**, representing the number of elements in the string, followed by the *newline* character, then the positive integer number **n**, representing the number of elements in each block, followed by the *newline* character. On the following **N** lines are the elements of the string (0 or 1), one per line, followed by the *newline* character.

Output data

The software will print on the first line the value of the compression ratio, with two decimals, rounded, then the compressed string, one number (0 or 1) per line.

ATTENTION to the compliance to the problem requirements: the display of results must be done EXACTLY as required! In other words, on the standard output stream there will be nothing displayed in addition to the problem requirements; following the automatic evaluation, any supplemental character displayed, or any display different than the requirements, will produce an eroneous result and will lead to the „Reject" of the solution.

Restrictions and remarks

1. 8< **N**<= 1024
2. 2 < **n** <= 8
3. All elements of the string are positive integers represented on 8 bits.
4. **Warning:** According to the chosen programming language, the file containing the code must have one of the extensions .c, .cpp, .java, or .m. The web editor does not add automatically these extensions and the lack of the extensions leads to the impossibility of program compilation!
5. **Warning:** The source file must be named by the candidate as: <name>.<ext> where name is the family name (last name) of the candidate and the extension is the one chosen according to the previous warning. Attention to the restrictions imposed by the Java language regarding the class name and the file name!

Examples

Input	**Output**	**Explanation**
8 2 1 0 0 0 0 0 1 1	0.89 0 1 1 0 0 0 1 1 1	The binary string has 8 bits and must be partitioned in blocks formed by 2 bits. The resulting blocks are: 10 00 00 11. For the coding that ignores the blocks composed only from bits of 0, the coding will transform the four blocks of the string as follows: 10 is coded as 110 (the block is copied and prefixed by a bit of 1) 00 is coded as 0 (the block is replaced by a single bit of 0) 00 is coded as 0 (the block is replaced by a single bit of 0) 11 is coded as 111 (the block is copied and prefixed by a bit of 1) The coded string will be 0 110 0 0 111, and contains 9 bits. The compression ratio is 8/9 For the coding that ignores the blocks composed only from bits of 1, the coding will transform the four blocks of the string as follows: is coded as 010 (the block is copied and prefixed by a bit of 0) 00 is coded as 000 (the block is copied and prefixed by a bit of 0) 00 is coded as 000 (the block is copied and prefixed by a bit of 0) 11 is coded as 1 (the block is replaced by a single bit of 1) The coded string will be 1 010 000 000 1, and contains 11 bits. The compression ratio is 8/11. The biggest compression ratio is obtianed for the coding that ignores the blocks composed only by 0 bits, thus the output string is 011000111, and the compression ratio is 8/9 = 0.89
10 5	1.25 0	The binary string has 10 bits and must be partitioned in blocks formed by 5 bits. The resulting blocks are: 00000 00011

Input	Output	Explanation
0 0 0 0 0 0 0 0 1 1	0 1 0 0 0 1 1	For the coding that ignores the blocks composed only from bits of 0, the coding will transform the four blocks of the string as follows: 00000 is coded as 0 (the block is replaced by a single bit of 0) 00011 is coded as 100011 (the block is copied and prefixed by a bit of 1) The coded string will be 0 0 100011, and contains 8 bits. The compression ratio is 10/8. Pentru codarea bazată pe ignorarea blocurilor de 1, codarea va transforma cele două blocuri din şir după cum urmează: 00000 is coded as 000000 (the block is copied and prefixed by a bit of 0) 11111 is coded as 1 (the block is replaced by a single bit of 1) The coded string will be 1 000000 1, and contains 8 bits. The compression ratio is 10/8. Both compression ratios are the same, so the output will be, as required, the coding that ignores the blocks composed only by 0 bits, thus the output string is 00100011, and the compression ratio is 10/8= 1.25
10 3 1 1 1 0 0 0 1 1 1 0	1.11 1 1 0 0 0 0 1 0 0	The binary string has 10 bits and must be partitioned in blocks formed by 3 bits. The resulting blocks are: 111 000 111 and the incomplete block 1 For the coding that ignores the blocks composed only from bits of 0, the coding will transform the four blocks (three complete and one incomplete) of the string as follows: 111 is coded as 1111 (the block is copied and prefixed by a bit of 1) 000 is coded as 0 (the block is replaced by a single bit of 0) 111 is coded as 1111 (the block is copied and prefixed by a bit of 1) 0 is coded as 10 (the incomplete block is copied and prefixed by a bit of 1) The coded string will be 0 1111 0 1111 10, and contains 11 bits. The compression ratio is 10/11. For the coding that ignores the blocks composed only from bits of 1, the coding will transform the four blocks (three complete and one incomplete) of the string as follows: 111 is coded as 1 (the block is replaced by a single bit of 1) 000 is coded as 0000 (the block is copied and prefixed by a bit of 0) 111 is coded as 1 (the block is replaced by a single bit of 1) 0 is coded as 00 (the incomplete block is copied and prefixed by a bit of 0) The coded string will be 1 1 0000 1 00, and contains 9 bits. The compression ratio is 10/9. The biggest compression ratio is obtianed for the coding that ignores the blocks composed only by 1 bits, thus the output string is 110000100, and the compression ratio is 10/9= 1.11

Worktime: 120 minutes