Problem G
MrCodeFormatGrader

Instructor Bob is building an automatic source code formatting grading website. The name of the website is "MrCodeFormatGrader". When you run Bob's program and submit your source code to it, the program will output C, the number of lines of source code; N, the number of source code lines that have format errors; and a list of line numbers that are not properly formatted. The line numbers are separated by spaces and in increasing order.

So for a 100-line program, with 10 errors occurring on lines 2, 3, 5, 10, 11, 12, 25, 26, 88, and 89, the original program might output this:

```
100 10
2 3 5 10 11 12 25 26 88 89
```

However, the people that used this program said that this output is not very readable. Your job is to read the output above and show two lists: a compressed list of errors and another compressed list of correct lines.

Input

Input consists of two lines. The first line contains two integers: C, the number of lines of source code in the program being checked, where $2 \leq C \leq 100\,000$; and N, the number of source code lines containing format errors, where $1 \leq N \leq \min(C, 1\,000)$. The second line of input will contain N values, identifying the line numbers of source code lines with format errors. These values are positive integers less than or equal to C and are listed in strictly increasing order.

Output

Output consists of two lines. The first line should start with the string: "`Errors:`", followed by a space, followed by a list of lines with errors. The second line of output should start with the string: "`Correct:`", followed by a space, followed by a list of lines without errors. For both lists, a contiguous set of line numbers should be represented by listing the first and last line, separated by a hyphen ('-'). In lists with two or more items, items should be separated by a comma and space except for the last value (or last range of contiguous values) of each list, which should be separated with an "and" instead of a comma and space. See the sample output. Note: every program will have at least one error and at least 1 correct line.

Sample Input 1

```
100 10
2 3 5 10 11 12 25 26 88 89
```

Sample Output 1

```
Errors: 2-3, 5, 10-12, 25-26 and 88-89
Correct: 1, 4, 6-9, 13-24, 27-87 and 90-100
```

Sample Input 2

```
40 18
1 3 4 6 7 8 9 12 13 14 20 25 26 27 28 30 38 40
```

Sample Output 2

```
Errors: 1, 3-4, 6-9, 12-14, 20, 25-28, 30, 38 and 40
Correct: 2, 5, 10-11, 15-19, 21-24, 29, 31-37 and 39
```