Course Assignment 2020

You are asked to develop a C program that will consult the user on what numbers to choose, in order to participate in Lotto draws. The consulted numbers have to be based on Lotto draw records and specific metrics that result after statistical analysis on these records.

LOTTO

In its simplest form (1 simple column), the player selects 6 numbers on scale of 1 to 49. On a regular basis (usually twice a week), there are scheduled draws, from which emerge 6 basic numbers + 1 additional one. In case that within the numbers chosen by player there are at least 3 out of 6 that match with the basic ones, then the player's column is considered as a winning column and the player can claim a prize, depending on the exact multitude of matching numbers. In particular, there are 4 basic winning categories:

- Category V: 3 correct numbers
- Category IV: 4 correct numbers
- Category III: 5 correct numbers
- Category I: 6 correct numbers

There is one more, special category:

- Category II: 5+1 correct numbers. In this category, the player has correctly guessed 5 out of 6 basic numbers as well as the additional number.

A player has the right to purchase a ticket for selecting more than 6 numbers, e.g., 8 numbers, increasing, this way, their chances to win a prize. However, in such a case, the participation fee increases (for example a ticket for 8 numbers corresponds to 28 simple columns, so, given the ticket price for the simple column (let it be 0.5€ for 6 numbers), the participation fee for 8 numbers would be 14€).

Example

Let's suppose that Bob and Alice would like to participate in a Lotto draw. Bob purchases a simple ticket (6 numbers), while Alice purchases two tickets, as shown in the following table:

	Selected numbers	**Ticket columns**	**Indicative participation fee**
Bob	2, 3, 5, 21, 38, 42	1	0.5€
Alice (1st ticket)	1, 4, 13, 19, 40, 49	1	1€
Alice (2nd ticket)	1, 3, 4, 5, 13, 19, 21, 42, 49	252	126€

Now, for the shake of example, let's suppose that the drawn numbers are 13, 42, 38, 5, 19, 3 (basic numbers) + 21 (additional number). Then:

A. Bob's ticket has 4 correct numbers (3, 5, 38, 42), so it belongs to the winning category IV.

B. Alice's 1st ticket has just 2 correct numbers (13, 19), so it does not belong to any of the winning categories

C. Alice's 2nd ticket has 5+1 correct numbers (3, 5, 13, 19, 42 + 21), so it belongs to the winning category II.

Back to the actual problem you are asked to solve, suppose that the following table represents the available draw records:

DRAW	**DATE**	**DRAWN NUMBERS**						
		1st	**2nd**	**3rd**	**4th**	**5th**	**6th**	**7th**
2171	26/12/2020	8	30	39	46	26	3	12
2170	23/12/2020	45	17	7	13	39	15	47
2169	19/12/2020	44	12	2	21	28	26	38

2168	16/12/2020	31	12	5	24	29	4	32
2167	12/12/2020	23	34	2	12	38	10	5
2166	09/12/2020	27	3	25	36	6	11	44
2165	05/12/2020	29	1	8	26	6	28	4
2164	02/12/2020	38	27	17	6	16	18	44
2163	28/11/2020	48	32	23	1	3	24	27
2162	25/11/2020	24	46	16	41	10	11	5

*Draw records "**last.txt**"*

Based on these records, we can easily calculate the following metrics:

1. Metric 0: Frequency of appearance of number *n*, ***f(n)***: How many times has number n been drawn in the *K* draws that are stored on the file.
2. Metric 1: Delay of appearance of number *n*, ***d(n)***: Before how many draws has number *n* appeared for the last time. In case an arbitrary number *m* has been drawn in the most recent draw, then *d(m)=0*, whilst if number *m* has not been drawn at all in the last *K* draws we analyze, then *d(m)=K*.
3. Metric 2: ***f(n)+d(n)***
4. Metric 3: Relative delay of appearance of number *n*. ***rd(n) = d(n) – K/f(n)***: The factor *K/f(n)* represents the mean delay of the number; thus, the mean number of draws that intervene between two consecutive appearances of number *n*. For example, number 6 has been drawn 3 times in the last 10 draws, thus its mean delay is 3.33 (10/3). As a result, *d(6)=5*, since the last appearance of number 6 was 5 draws ago. So, *rd(6) = 5 – 3.33 = 1.67*, which means that, statistically speaking, number 6 was expected to be drawn 1.67 draws before. In case that a number has not been drawn in the last *K* draws available, then *K/f(n)* is set to *K*, so *rd(n)=0*.

Examples

Number	f(n)	d(n)	f(n) + d(n)	rd(d)
12	4	0	4	-2,5
11	2	5	7	0
20	0	10	10	0
34	1	4	5	-6
16	2	7	9	2

Tasks to complete

Develop a C program that gets as input from the user 2 integers (e.g. numbers and metric), where (0 < *numbers* < 50) and (-1 < *metric* < 4) as well as a string (e.g. *filename*) that represents the name of the data file that contains the draw history (each one of the file lines represents a separate draw and contains 7 integers withing the range [1, 49] separated by space). In case at least one of the integers is out of bounds or the file name does not exist, the program should print a respective error message to the console and then terminate. Else, the program has to print to the console the *numbers* best numbers, according to the metric *serial number* (higher metric value of a number == better number). If there are multiple numbers with the same metric value, the program has to select the shorter ones (e.g. if numbers 45 and 7 have the same metric value, the program must select number 7). The numbers have to be printed in descending order, according to metric *metric*.

Execution Examples

Input	Output
60 0 last.txt	Wrong Input!
5 4 last.txt	Wrong Input!
5 3 this_file_does_not_exist	File Error!
5 0 last.txt	12 3 5 6 24
6 1 last.txt	9 14 19 20 22 33 35 37 40 41
10 2 last.txt	16 6 27 1 9 11 14 19 20 22 33 35 37 40 42 43 49 5 24 10

Indicative data file

```
 8   30  39  46  26  3  12
45   17  7  13  39  15  47
44   12  2  21  28  26  38
31   12  5  24  29  4  32
23   34  2  12  38  10  5
27  3   25  36  6  11  44
 29  1   8   26  6  28  4
```

*Data file "**last.txt**" of the previous examples*