The main goal of this program is to convert a given number from one base to another base. The input to this program is given/fed **ONLY** through a file and not using the terminal to manually type in input. The semantics as to how the program reads the input file is up to you i.e., the program can ask the user for the name of the input file or the program can assume that there is an `input.txt` file present in the same directory as the source code (suggested method).
The input file will contain one or more lines, and each line in the input file has the following format:
`<number> <currentBase> <BasetoConvertTo> <MaxNumberOfBits>`.
Each of these 4 values are seperated by a single space.

## Sample Input

```
256 7 4 10
1234 10 2 3
24756 8 16 7
```

Your program should process one line at a time. The first line in the sample input implies that the number `256` is of base `7` and needs to be converted to base `4`. The resulting value of converting 256 from base `7` to base 4 should be shown in 10 digits. If the resulting value after conversion is than 10 digits, append zeros at the beginning to make the result consist of 10 digits. Alternatively, if the resulting value of converting a number needs more than 10 digits, return `OVERFLOW`. The second line of the sample input file demonstrates this example. Converting 1234 from decimal to binary will result in 10011010010, which is 11 digits. Hence, the output should be `OVERFLOW`.

The output of the program should **NOT** be displayed on the terminal, instead the output of processing the entire input file must be written to another file line by line into `output.txt`, as shown in the sample output.

## Sample Output

```
0000002023
OVERFLOW
00029EE
```

## Program Structure

**Your program should not use inbuilt python libraries to convert from one base to another.**
The source code should contain at the very least a class named `Number`. The `Number` class will house the following subroutines at the bare minimum:

- A subroutine to convert a number from base '10' to base 'r' and returns the result of the converted value.
- A subroutine that converts a number from base 'r' to base '10' and returns the result of the converted value.
- (Optional) A constructor to manage the numbers as objects.
- (Optional) If needed, other subroutines can be added to the class as well.

## Tips

- Comment your source code appropriately, include an informative header at the top of your program, and use good coding style.
- Use meaningful variable names.

- Consider the different possible end cases in your program.
- Output of the program **must** be written to another file and not displayed on the terminal.
- Of course, feel free to Google things to help you. As always, do not Google solutions and make sure to *cite sources*.
- The source code need not be written in Python, feel free to use any object oriented language - C++/C#/Java/Python.
- It is suggested that the class and main program are submitted in separate files.
- You can assume that there are no errors in the input files.
- A minimum of 30% points will be deducted for programs that contain compilation errors.

## Submission

Upload the source code containing the `Number` class file and the main program onto Canvas. Both the class and the main program can be present in the same file. In case of using a programming language other than Python, attach a simple Readme.txt file detailing the steps to run your program.

## Rubric

| Item | Points |
|---|---|
| Good coding style | 2 |
| Appropriate Comments & Header | 2 |
| Input read from file | 5 |
| Output written to file | 5 |
| Subroutine to convert from base 10 to base 'r' | 4 |
| Subroutine to convert from base 'r' to base 10 | 4 |
| Output is correct | 5 |
| Zeros are appending as necessary | 3 |
| Total | 30 |

FIN