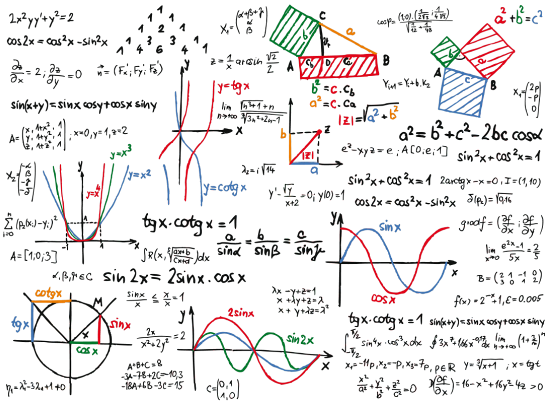


B2- Mathematics

B-MAT-200

107transfer

Polytechnical Polynomials

Transfer

Polytechnical Polynomials

binary name:	107transfer
repository name:	107transfer_$ACADEMICYEAR
repository rights:	ramassage-tek
language:	C, C++, perl 5, python 3 (≥ 3.5), ruby 2 (≥ 2.2), php 5.6, bash 4
group size:	1-2
compilation:	via Makefile, including re, clean and fclean rules

- Your repository must contain the totality of your source files, but no useless files (binary, temp files, obj files,...).
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

A laboratory performs tests on new electronic components to be integrated into its last generation chipset. Those components are entirely characterized by their transfer function, which determines frequency response ; this function processes the input frequency and computes an output frequency (caracterizing the way the component amplifies or reduces the input frequency). The transfer functions of these components are represented rational functions.

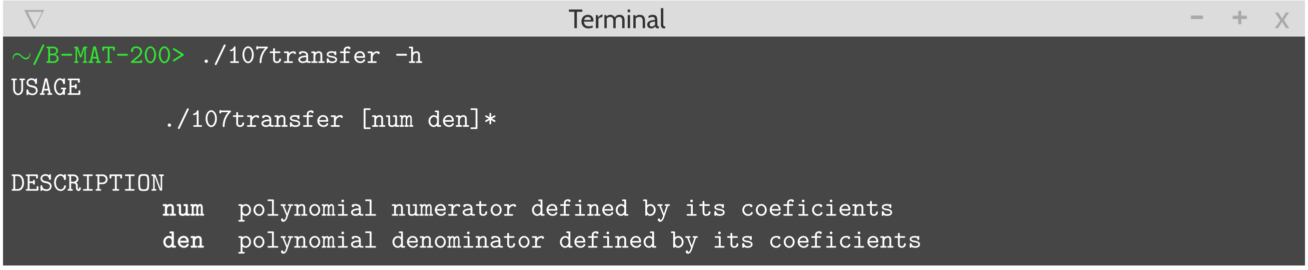
To execute the tests, the laboratory is used to using a quite complex software. Well, they used to : the software was so badly benchmarked, they decided to have a new one developped, and guess what, you are in charge of the optimization of the transfer function computations.

You have to manage several components in cascade. In such a case, the transfer function of the set of components is the product of the transfer functions of each component.

A transfer function is here defined by two strings (one for the numerator, one for the denominator), composed by the polynomial coeficients split by the '*' sign.
For instance, "1*4*2*6*0*8" stands for $8x^5 + 6x^3 + 2x^2 + 4x + 1$.

Your program has to print the frequency responses of the component for the values from 0 to 1, by 0.001 steps.

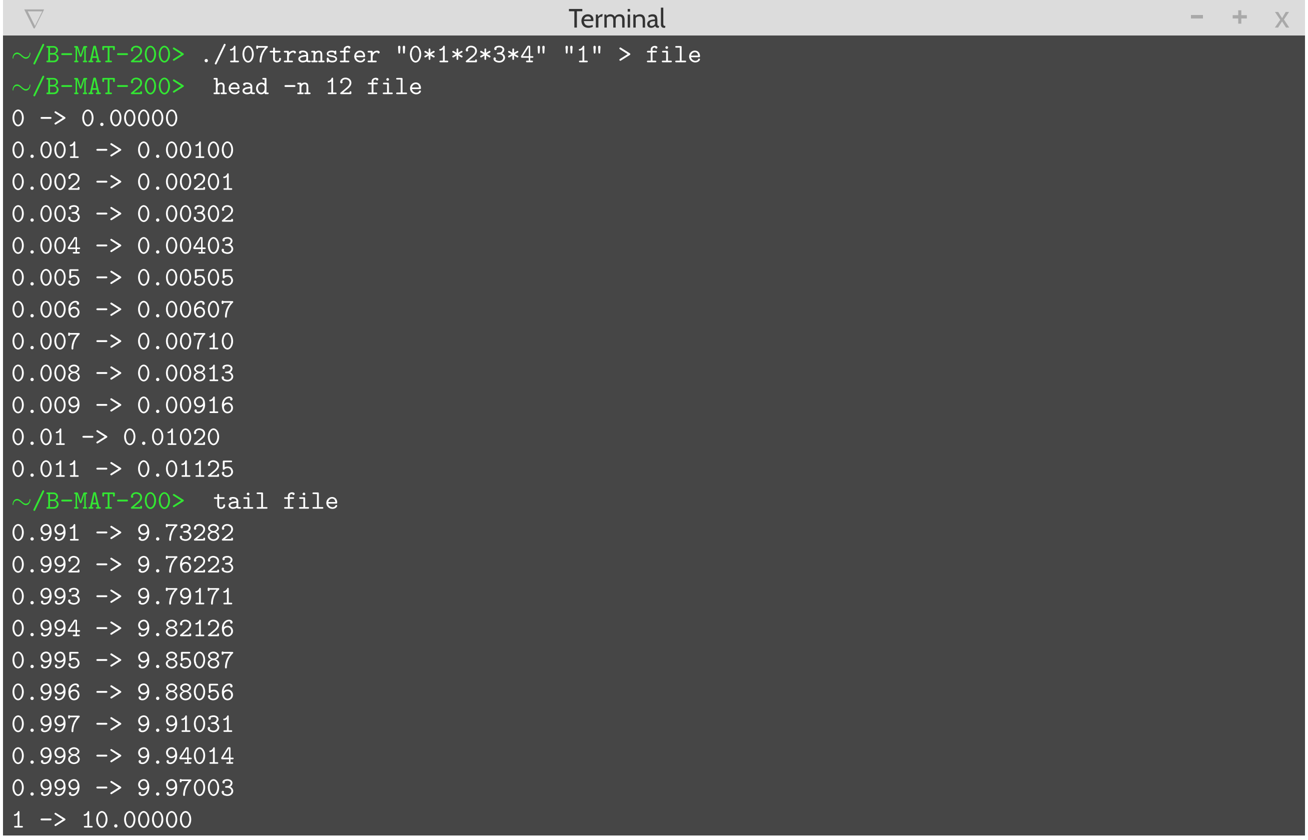
Considering the context, too slow programs will be considered non functional. You'd better use a fast algorithm...



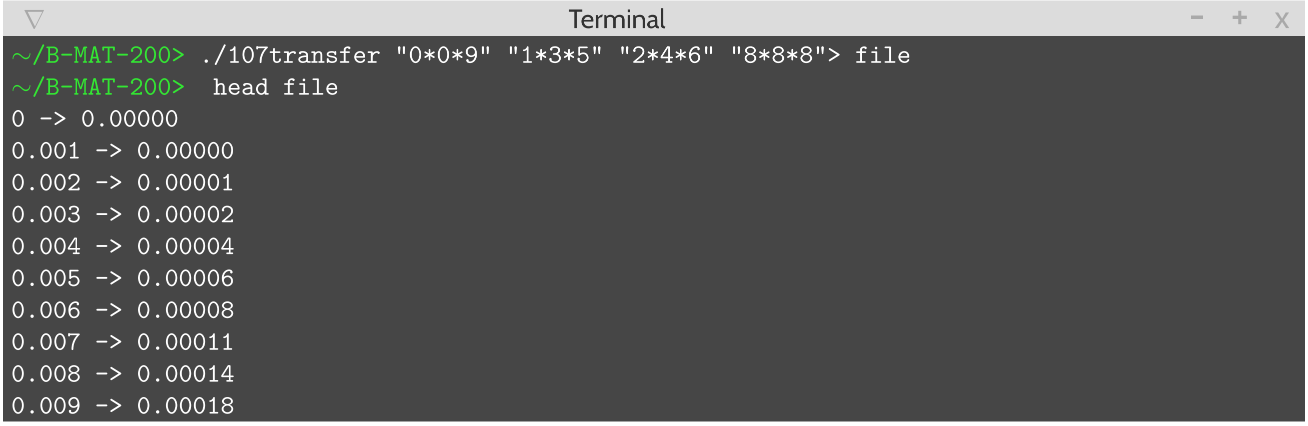
```
~/B-MAT-200> ./107transfer -h
USAGE
           ./107transfer [num den]*

DESCRIPTION
           num   polynomial numerator defined by its coeficients
           den   polynomial denominator defined by its coeficients
```

Your program output has to be strictly identical to the one below.



```
~/B-MAT-200> ./107transfer "0*1*2*3*4" "1" > file
~/B-MAT-200>  head -n 12 file
0 -> 0.00000
0.001 -> 0.00100
0.002 -> 0.00201
0.003 -> 0.00302
0.004 -> 0.00403
0.005 -> 0.00505
0.006 -> 0.00607
0.007 -> 0.00710
0.008 -> 0.00813
0.009 -> 0.00916
0.01 -> 0.01020
0.011 -> 0.01125
~/B-MAT-200>  tail file
0.991 -> 9.73282
0.992 -> 9.76223
0.993 -> 9.79171
0.994 -> 9.82126
0.995 -> 9.85087
0.996 -> 9.88056
0.997 -> 9.91031
0.998 -> 9.94014
0.999 -> 9.97003
1 -> 10.00000
```



```
∼/B-MAT-200> ./107transfer "0*0*9" "1*3*5" "2*4*6" "8*8*8"> file
∼/B-MAT-200>  head file
0 -> 0.00000
0.001 -> 0.00000
0.002 -> 0.00001
0.003 -> 0.00002
0.004 -> 0.00004
0.005 -> 0.00006
0.006 -> 0.00008
0.007 -> 0.00011
0.008 -> 0.00014
0.009 -> 0.00018
```