Write a complete C program that

- takes 2 numbers as strings,
- convert them into integers,
- calculate the combination of them,
- save each value in a struct and,
- print the combination as an output.

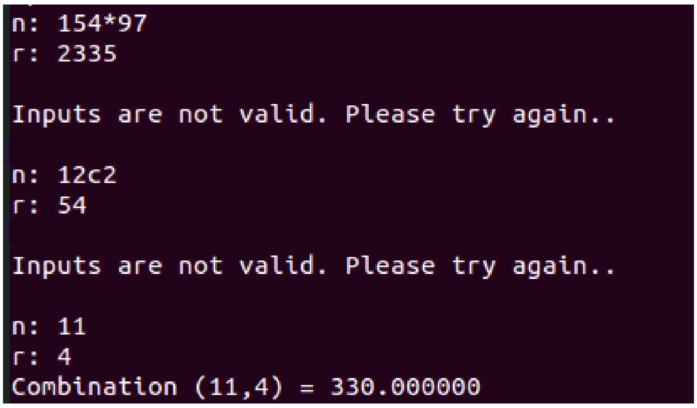
You should use 2 different structs:

1) First one is called 'struct number' with 2 variables: a string of the number and the integer. Both keeps the same number. The string is the input coming from the user. The integer should be calculated via a recursive function (parse_to_int).
   Sample values for the variables of the struct:
   string : "120"
   int: 120
2) The second struct is called 'struct combination'. It has three variables: *n, r* and *result. n* and *r* are instances of *'struct number'* and *result* is an integer. In other words, to save n and r, you should refer to the *'struct number'*, you can't save them directly in *'struct combination'*.

Functions needed:

1) int factorial (int n)
   A recursive function which calculates end returns the factorial of an integer n.
2) int is_number (char *number_as_str, int length_of_str)
   A recursive function to check if the input is proper (if it is a number that can be converted to an int).
   For example, "1223" is a proper input while "123b2" is not.
   The function should return 1 if the string is a convertible number, 0 otherwise.
   If an input is not a convertible number, the program should ask for another input.
3) int parse_to_int (char *number_as_str, int length_of_str)
   The recursive function which converts a given string to int, and then returns it.
4) int str_len(char *str)
   To use in 2nd and 3rd functions, write a recursive function to find the length of a given string.

Expected output of the program:



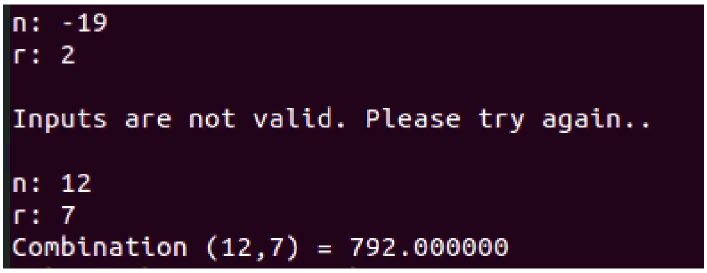
```
n: 154*97
r: 2335

Inputs are not valid. Please try again..

n: 12c2
r: 54

Inputs are not valid. Please try again..

n: 11
r: 4
Combination (11,4) = 330.000000
```



```
n: -19
r: 2

Inputs are not valid. Please try again..

n: 12
r: 7
Combination (12,7) = 792.000000
```

Notes:

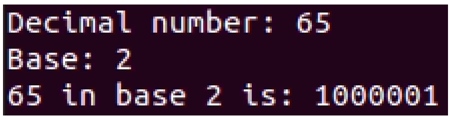
- You can’t use a function from an existing library to implement given tasks.
- Make sure you save every value (both inputs and output) in structs. You can’t keep them in local variables where you read the input / where you make the calculations.
- Combination formula:

$$C(n,k) = \binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

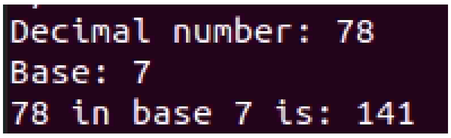
Write a complete C program that reads a decimal number and a base value (both as integers) and implements base conversion by using a recursive function.

Function prototype: int base_converter (int decimal_number, int base)

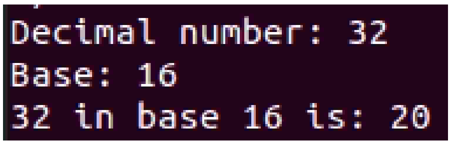
Expected output:



```
Decimal number: 65
Base: 2
65 in base 2 is: 1000001
```



```
Decimal number: 78
Base: 7
78 in base 7 is: 141
```



```
Decimal number: 32
Base: 16
32 in base 16 is: 20
```