

**Topic: Static and Shared Libraries**

Two countries of a planet in the FB23 galaxy uses two different conventions for spelling large numbers (positive integers). Here, *spelling* a number means writing the number in words.

**FooSA** uses units, tens, hundreds, thousands, millions ( $10^6$ ), billions ( $10^9$ ), trillions ( $10^{12}$ ), and so on.

**Barat** uses units, tens, hundreds, thousands, lakhs ( $10^5$ ), crores ( $10^7$ ), and so on.

For example, in the foo system, the number 1234567890 is written as *one billion two hundred thirty four million five hundred sixty seven thousand eight hundred ninety*. The same number in the bar system is written as *one hundred twenty three crore forty five lakh sixty seven thousand eight hundred ninety*. Both foo and bar people write 0 as *zero*, otherwise they avoid using zero in a spelling. For example, the spellings of  $10^8 + 9$  in the foo and bar systems are *one hundred million nine* and *ten crore nine*, respectively.

In this assignment, we restrict the universe of numbers to 32-bit unsigned int values, so the largest number you deal with will be  $2^{32} - 1 = 4294967295$ . Your task is to develop static and shared libraries for writing such numbers in words, in both the foo system and the bar system. Name your library as **numprn**. You should write your code in plane C, invoke the gcc compiler, and use no global variables (or arrays).

**Part 1: Prepare the files**

Create a subdirectory **A1** in your home directory, and work in that subdirectory, Prepare the following files using a text editor. *Do not use any file name other than what is suggested below.*

Use the **#include** ". . ." format for including (non-system) header files.

The Library Files: The **numprn** library consists of three parts.

**common.h and common.c** Both the foo and the bar systems write the numbers in the range 0–999 in the same manner. These numbers are needed for printing larger numbers as in *two hundred thirty four million* and *one hundred twenty three crore*. Write a function **cmnprn()** that takes an unsigned int *n* as its only parameter. If *n* is in the range 0–999, the function prints *n* in words without any leading or trailing spaces or newline characters. You are not supposed to call this function for  $n > 999$ . But if you do so, it prints *big*.

**foo.h and foo.c** These are for printing any number in the range 0 to  $2^{32} - 1$  in the foo system. Write a function **fooprn()** that takes an unsigned int *n* as its only parameter, and prints *n* in words in the foo system, without any leading or trailing spaces or newline characters.

**bar.h and bar.c** These are for printing any number in the range 0 to  $2^{32} - 1$  in the bar system. Write a function **barprn()** that takes an unsigned int *n* as its only parameter, and prints *n* in words in the bar system, without any leading or trailing spaces or newline characters.

All the three functions mentioned above should return **void**. The library does not require any new data type, so the header files would consist only of the function prototypes. You should not use any global variable anywhere in your codes.

The Application Program: Write a program **app.c** to do the following. A compilation-time macro flag **INTERACTIVE** instructs the program to run in interactive or non-interactive mode. In the interactive mode, the program reads an unsigned int *n* from the user. In the non-interactive mode, the program generates a random unsigned int *n*. The program enters a loop that keeps on printing *n* in both the foo and the bar formats, and then dividing *n* by 10. The loop stops after *n* reduces to 0 (it should print 0 in the last iteration). The loop calls the library functions **fooprn()** and **barprn()** for the printing purpose. It additionally prints a double quote immediately before and a double quote immediately after the printing made by each of these library calls. See the sample output below.

## Part 2: Build the libraries, and run the application program

Prepare a static `numprn` library from the `common`, `foo`, and `bar` files mentioned above. Compile `all.c` using the static library in the two modes (interactive and non-interactive), and verify the correctness of your library and application program.

Prepare a shared (dynamic) `numprn` library from the `common`, `foo`, and `bar` files mentioned above. Compile `all.c` using the shared library in the two modes (interactive and non-interactive), and verify the correctness of your library and application program.

Also, prepare a text file `README.txt` which mentions the commands you use to do the following.

- (i) Build the static library.
- (ii) Compile and run the application program using the static library.
- (iii) Build the shared library.
- (iv) Compile and run the application program using the shared library.

## Part 3: What you submit

Delete all the files except those that you have written (the `.c` and `.h` files and `README.txt`). Go to the parent directory. Prepare a single archive in one of the following two ways.

```
zip -r A1.zip A1/
```

```
tar cvzf A1.tgz A1/
```

Submit `A1.zip` or `A1.tgz`.

## Sample output

```
n = 4097612392
foo: "four billion ninety seven million six hundred twelve thousand three hundred ninety two"
bar: "four hundred nine crore seventy six lakh twelve thousand three hundred ninety two"

n = 409761239
foo: "four hundred nine million seven hundred sixty one thousand two hundred thirty nine"
bar: "forty crore ninety seven lakh sixty one thousand two hundred thirty nine"

n = 40976123
foo: "forty million nine hundred seventy six thousand one hundred twenty three"
bar: "four crore nine lakh seventy six thousand one hundred twenty three"

n = 4097612
foo: "four million ninety seven thousand six hundred twelve"
bar: "forty lakh ninety seven thousand six hundred twelve"

n = 409761
foo: "four hundred nine thousand seven hundred sixty one"
bar: "four lakh nine thousand seven hundred sixty one"

n = 40976
foo: "forty thousand nine hundred seventy six"
bar: "forty thousand nine hundred seventy six"

n = 4097
foo: "four thousand ninety seven"
bar: "four thousand ninety seven"

n = 409
foo: "four hundred nine"
bar: "four hundred nine"

n = 40
foo: "forty"
bar: "forty"

n = 4
foo: "four"
bar: "four"

n = 0
foo: "zero"
bar: "zero"
```