Problem: Directory Size Analysis with Compact Input

Problem:

You are given a compact representation of a directory structure, where each directory can have files and subdirectories. The input format is as follows:

```
dir <directory_name> ( <file_size_1> <file_size_2> ... dir <subdirectory_name>( ... ) ... )
```

Output Format:

Output the size of each directory, including its files and nested subdirectories. The output format is:

<directory_name>: <directory_size>

Example:

Input:

```
dir documents ( 2048 512 ) dir photos ( 3072 dir summer ( 2048 1024 ) )
```

Output:

```
documents: 2560

photos: 6144

summer: 3072
```

Problem: Bits

Implement a function that takes in a uint32_t number and returns the maximum number that cand be obtained by rotating the number to the right.

Example:

Function input: 0x000F

Function return value: 0xF000

Problem: String replace

Implement a C function that will find, search and replace text strings. Receives three strings (s1, s2 and s3) and replaces all the occurrences of string s2 into the first string (s1), with the third string (s3), returning the new string. The replacement phrase can be shorter or longer than the searched for phrase. Show the updated initial line of text on the display.

Problem: Print numbers

Read line from a file given as a command line argument and print out the valid positive numbers found on the lines. A valid number is defined as a number separated by spaces (exception start and end of a line where the number is not separated by spaces).

Problem: Sort students

Read data from a file given as a command line argument. Every line contains input in the following format: “student-name: grade-average”. The file is technically a student database. Read this file and print out the students in decreasing order based on the grade-average. Break ties by alphabetically sorting the students.