Practice Arena

Practice problems aimed to improve your coding skills.

- PRACTICE-02_SCAN-PRINT
- PRACTICE-03_TYPES
- LAB-PRAC-02_SCAN-PRINT
- LAB-PRAC-01
- PRACTICE-04_COND
- BONUS-PRAC-02
- LAB-PRAC-03_TYPES
- PRACTICE-05_COND-LOOPS
- LAB-PRAC-04_COND
- LAB-PRAC-05_CONDLOOPS
- PRACTICE-07_LOOPS-ARR
- LAB-PRAC-06_LOOPS
- LAB-PRAC-07_LOOPS-ARR
- LABEXAM-PRAC-01_MIDSEM
- PRACTICE-09_PTR-MAT
    - Monster Multiply Revisited
    - Proper Case
    - Num2Word
    - Mr C meets Matrices
- LAB-PRAC-08_ARR-STR
- PRACTICE-10_MAT-FUN
- LAB-PRAC-09_PTR-MAT
- LAB-PRAC-10_MAT-FUN
- PRACTICE-11_FUN-PTR
- LAB-PRAC-11_FUN-PTR
- LAB-PRAC-12_FUN-STRUC
- LABEXAM-PRAC-02_ENDSEM
- LAB-PRAC-13_STRUC-NUM
- LAB-PRAC-14_SORT-MISC

Num2Word

PRACTICE-09_PTR-MAT

A long variable takes 8 bytes to store whereas a char takes 1 byte. This means a long variable has enough space to store 8 characters inside it. In the input, you will be given a single non-negative integer. Extract the 8 characters stored inside that long integer and store them inside a character array with 9 characters - the 9th character should be set by you to the NULL character. Print the character array as a string.

Example:
'W' = 87 = 01010111
'O' = 79 = 01001111
'\0' = 0 = 00000000 (in binary)

So the binary number 01010111 01001111 01010111 00000000 00000000 00000000 00000000 00000000
(spaces between bytes are just for ease of reading) which corresponds to the long integer 6291342861971488768
actually represents the sequence of 8 characters 'W' 'O' 'W' '\0' '\0' '\0' '\0' '\0'

HINT: There are at least three ways to solve this problem - let us know on Piazza if you find more

1. Method 1: use remainder and divide trick to extract bytes out of the long number
2. Method 2: a convenient way of doing method 1 using bitshift operators (search online to learn more - we will do them in class later).
3. Method 3: use a char* pointer to access individual bytes of the long variable. The week 8 tutorial will cover how to perform arithmetic using pointers. This can be used to extract characters from a long integer. WARNING: if using this method, be careful that the Prutor server is a *Little Endian* machine (search online to learn more) which means it actually stores the bytes in the reverse order i.e. the long number 6291342861971488768 will be stored as 00000000 00000000 00000000 00000000 00000000 01010111 01001111 01010111

Start Solving! (/editor/practice/6161)