project[3]: Arrays

Due Thursday, 10/15/2015, 11:59:59pm

Project Goals

The goals of this project are to explore the use of one dimensional and two dimensional arrays.

Project Description

This project consists of two programs:

Program 1 (Scramble a sentence):

Write a program that scrambles a sentence by: 1) reversing the order of words in a sentence and 2) shifting all the letters in each word by two positions to the right. When shifting to the right, the letters wrap around to the beginning. This program should:

- prompt a user to enter a sentence ending with either a period, a question mark or an exclamation mark
- output the same sentence with all of the words, with letters shifted to the right by two positions and printed in reverse order.

Here is an example of how the program should behave:

```
Enter a sentence ended by a '.', a '?' or a '!': you can cage a
swallow can't you?

Reversed sentence: ouy 'tcan owswall a geca anc ouy?
```

Code for this program should be in a file named `scramble_sentence.c`

Program 2 (Magic square):

Write a program that prints an 9x9 magic square (a square arrangement of the numbers 1, 2, ..., n^2 in which the sums of the rows, columns and diagonals are all the same). Strategy: store the magic square in a two-dimensional array. Start by placing the number 1 in the middle of row 0. Place each of the remaining numbers 2, 3, ..., n^2 by moving up one row and over one column. Any attempt to go outside the bounds of the array should “wrap around” to the opposite side of the array. For example, instead of storing the next number in row-1, we would store it in row n-1 (the last row). Instead of storing the next number in column n, we would store it in column 0. If a particular array element is already occupied, put the number directly below the previously stored number.

For example, for a magic square of size 5, your program should print:

```
17   24    1    8   15
23    5    7   14   16
 4    6   13   20   22
10   12   19   21    3
11   18   25    2    9
```

Constraints

- ❏ You must **use a 2-dimensional array**.
- ❏ Make sure that your program outputs the numbers aligned on the right on the columns, similar to the example above

Code for this program should be in a file named `magic_square.c`

Program 3 - Challenge (Hexadecimal addition)

Write a program to perform addition of two hexadecimal numerals, each with up to 10 digits. If the result of the addition is more than 10 digits long, the program should output the message “Addition overflow” and not the result of the addition.

This program should:

- ❏ prompt a user to enter the two numbers to be added
- ❏ output sum of the two numbers or the message “Addition overflow”.

Here is an example on how the algorithm should behave:

```
Please enter the first hexadecimal number (10 digits maximum): 13A570
Please enter the second hexadecimal number (10 digits maximum): 3CA21
The sum of the two numbers is: 176F91
```

Note: the numbers can have any number of digits between 1 and 10.

Constraints

- ❏ You must use **arrays** to store hexadecimal numerals as arrays of characters.

Code for this program should be in a file named `add_hex.c`

Submission details

The project needs to be submitted before midnight, Thursday, 10/15/2015, 11:59:59pm.

To submit your project:

- create a directory called “project3”
- put ALL of your code in that directory
- save your description file into that directory
- DO THIS ONCE: Install the submission script ***(don’t type the ‘>’ symbols)***

  ```
  > cd ~
  > rm submit
  > wget http://www.cse.unr.edu/~newellz2/submit
  > chmod +x ./submit
  ```
- TO Submit:

  ```
  > cd project3
  > ~/submit
  ```

please use cs135 as the class name and project3 as the project name (no capitals).

The submission script copies all files in the current directory to our directory. You may submit as many times as you like before the deadline, we only keep the last submission.

Academic Honesty

Academic dishonesty is against university as well as the system community standards. Academic dishonesty includes, but is not limited to, the following:

Plagiarism: defined as submitting the language, ideas, thoughts or work of another as one's own; or assisting in the act of plagiarism by allowing one's work to be used in this fashion.

Cheating: defined as (1) obtaining or providing unauthorized information during an examination through verbal, visual or unauthorized use of books, notes, text and other materials; (2) obtaining or providing information concerning all or part of an examination prior to that examination; (3) taking an examination for another student, or arranging for another person to take an exam in one's place; (4) altering or changing test answers after submittal for grading, grades after grades have been awarded, or other academic records once these are official.

Cheating, plagiarism or otherwise obtaining grades under false pretenses” constitute academic dishonesty according to the code of this university. Academic dishonesty will not be tolerated and penalties can include canceling a student’s enrollment without a grade, giving an F for the course, or for the assignment. For more details, see the University of Nevada, Reno General Catalog.