Computer Programming Language

【Fall, 2018】

Homework 6

Program A： Bridge card game (50%)

In a typical card game, each player gets a hand of cards. The deck is shuffled and cards are dealt one at a time from the deck and added to the players' hands. Bridge uses a pack of 52 playing cards. There are 4 suits (spades, hearts, diamonds, clubs) each of 13 cards: 1 (the Ace) to 10 and Jack, Queen, King. The Ace is the highest card, followed by the King, Queen, down to the 2.

You are requested to design a program to simulate dealing cards to 4 players. The program needs to simulate three times of card shuffling and dealing, and saves the results into a file named CardGame.txt as well as displaying the results on the screen. Your program should also calculate and show the total high-card points of each player. High-card points are counted as follows: A = 4 points, K = 3 points, Q = 2 points, and J = 1 point. Design your own format for the program output.

■ *Web-Cat Submission Check:*

    int answer1;   // Store the high-card points of the first player in the final run of dealing.

    int answer2;   // Store the high-card points of the second player in the final run of dealing.

    int answer3;   // Store the high-card points of the third player in the final run of dealing.

    int answer4;   // Store the high-card points of the fourth player in the final run of dealing.

Program B： Morse code (50%)

Perhaps the most famous of all coding schemes is the Morse code, developed by Samuel Morse in 1832 for use with the telegraph system. The Morse code assigns a series of dots and dashes to each letter of the alphabet, each digit and a few special characters (such as period, comma, colon and semicolon). In sound-oriented systems, the dot represents a short sound, and the dash represents a long sound. Other representations of dots and dashes are used with light-oriented systems and signal-flag systems. In a sound-oriented system, a space is indicated by a short period of time during which no sound is transmitted. The international version of the Morse code is shown in the following table.

Write a program that reads a file named MorseCode.txt and convert it into Morse code. Use one blank between each Morse-coded letter and three blanks between each Morse-coded word. Display the result on the screen and save the converted Morse code into a file named MorseCode.dat. The file MorseCode.txt has the content as below

Morse code is one of several adapted computer access methods and alternative communication techniques that can be effective in helping enhance the lives of persons with special needs. Those without the ability to speak, sign, or use keyboards may benefit from this proven system of communication.

Morse Code Alphabet
The International morse code characters:

A .-	N -.	0 -----
B -...	O ---	1 .----
C -.-.	P .--.	2 ..---
D -..	Q --.-	3 ...--
E .	R .-.	4 ....-
F ..-.	S ...	5 .....
G --.	T -	6 -....
H	U ..-	7 --...
I ..	V ...-	8 ---..
J .---	W .--	9 ----.
K -.-	X -..-	Fullstop .-.-.-
L .-..	Y -.--	Comma --..--
M --	Z --..	Query ..--..

Some information about Morse code can be found in the following link:

http://morsecode.scphillips.com/

■ *Web-Cat Submission Check:*

    int answer1;   // Store the total number of words in MorseCode.txt
    int answer2;   // Store the total number of characters (excluding comma, period, and space) in
                    MorseCode.txt

Challenge Program: Pangram Checker (Bonus Points 40%)

A pangram or holoalphabetic sentence is a sentence using every letter of a given alphabet at least once. Pangrams have been used to display typefaces, test equipment, and develop skills in handwriting, calligraphy, and keyboarding. The best known English pangram is "The quick brown fox jumps over the lazy dog." It has been used since at least the late 19th century, was utilized to test Telex data communication equipment for accuracy and reliability, and is now used by a number of computer programs to display computer fonts.

Design a function PangramCheck to determine whether an input string sentence is a pangram, and write a main program to call the PangramCheck function to check the sentences in the file PangramList.txt. Find out the number of sentences which is not a pangram in the file and show these sentences on the screen.

■ *Web-Cat Submission Check:*

    int answer1;   // Store the total number of sentences which is NOT a pangram in the file.
    int answer2;   // Store the total number of sentences which is a pangram in the file.

Notes:

1. Please submit your programs (source codes) to the Web-CAT grading system website (http://140.112.94.129:8080/Web-CAT_1.4.0/WebObjects/Web-CAT.woa/) before **Dec. 13**. (3:30PM)

2. Late submission will have a penalty of 10% discount per day of your grade toward a minimum score of 60. No late submission over a week will be accepted.

3. Criteria of grading include: (1) Program functionality; (2). User interface; (3). Structure of the program; (4). Suitable comments; (5). Programming style; (6). Creativity. The proper use of functions in building a modular program is encouraged and will be part of the grading criteria in this homework.