



Computer Programming Language

【Fall, 2021】

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: Ace = 4 points, King = 3 points, Queen = 2 points, and Jack = 1 point. Design your own format for the program output.

■ *AUTOLAB Submission Check:*

```
int answer1; // Store the total high-card points of the first player in the final run of dealing.  
int answer2; // Store the total high-card points of the second player in the final run of dealing.  
int answer3; // Store the total high-card points of the third player in the final run of dealing.  
int answer4; // Store the total high-card points of the fourth player in the final run of dealing.
```

Program B : Scrabble Word Solver (50%)

In this exercise, you are asked to make your own Scrabble Word Solver program. To this program, you will simply input a string of letters and the program will print out (to the screen and to a file) a list of all the possible words you can make from a dictionary. For example, if you input a string “binou” of 5 characters, you will find the list as follows:

“bin”, “bio”, “bun”, “ion”, “nib”, “nob”, “nub”, “obi”, “uni”, “bi”, “in”, “no”, “nu”, “on”, “i”, “u”

To find the valid words, you have to specify a text file of words provided with this homework (**Dictionary.txt**). This file contains a fairly complete list of English words. After finding a list of valid words, you need to sort the list of valid words along with their number of characters in each word in descending order. You are requested to read the words from the dictionary file and store them in a list of string using dynamic memory allocation approach (pointer or vector (STL)) for further searching of valid words. Show the result of the valid words on the screen as well as save them in a text file named **ValidWords.txt**.



You need to design your program with a modular approach using functions. At least, but not limited to, two functions need to be implemented as specified below:

checkWord(string dictionaryWord, string myLetters) : Check that if the string **dictionaryWord** can be formed from the letters in the string **myLetters**. It is possible that **dictionaryWord** searched may contain multiples of the same letters. In this case, it is not a valid word.

outputValidWord(string *validWordList, string outFileName) : Output the list of the valid words to an output file name **outFileName**. (you may also use string array for **validWordList**)

■ *AUTOLAB Submission Check:*

```
int answer1;    // Store the total number of words in the Dictionary.txt file
int answer2;    // Store the total number of valid words if the input string is "stable"
```

Challenge Program: Pangram Checker (Bonus Points 25%)

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.

■ *AUTOLAB 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 AUTOLAB grading system website (<https://bblab.bime.ntu.edu.tw/>) before **Dec. 16 (2:20PM)**
2. Late submission will have a penalty of 10% discount per day of your homework total score toward a maximum of 50% discount. No late submission over five days 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.