BLG 233 – HOMEWORK 3 REPORT

Student Name: Halit Uyanık

Student No: 150140138

# 1-Main files and their insides

The fortune telling program includes 1 cpp and 5 header files. Header files and their respective objectives are:

FileOps.h

This header is used to define a struct to open a file easily.

Card.h

This header includes the most basic struct of the program. It defines the card structure which includes number, type, and a pointer for the next card that may be placed on it.

Queue.h

This header is used for queue structs. It mainly defines the printing, enqueue, dequeue and isEmpty functions for queue operations. It also has a front and back pointers for the queue struct.

Temporary.h

This header is used for stack structs. It includes push and pop functions, and a head pointer to hold stack struct elements.

Deck.h

This header holds and defines the most important aspects of our program. The struct deck defined inside includes an array of 13 integers for 13 type of cards. A char array of 4 elements for the 4 types of cards, 2 queue structs for main card queue and fortune cards, one temporary struct for stack structure, fileoperation for output file. It also initializes evaluateDeck, putCard, calculateFortune, initialize, and shuffleDeck functions. These functions will be explained in the following section.

Main.cpp

This file is used to initialize a deck and call the function to evaluate fortune.

# 2-How program Works

First we initialize a new deck struct. Then evaluateDeck function of deck is called. evaluateDeck function first calls srand in order to provide a different result for the fortune output each time it is called. Then program initializes the deck. At initialization process first a file is created for output. Then 2 queue is created for fortune cards and main deck, and a stack is created for the cards which will be put to temporary place. Finally shuffleDeck function is called with its int parameter as 0.

# Recursive Part

Shuffle process works as recursive function and initializes the main deck as queue structure. Recursive stops when the input parameter of function is equal to 52 which is the maximum number of cards that a deck can have in our program. If the number is not equal to 52 it takes a random number for 0 to 12 and checks the cardCounts array which has the information of how many cards from each 13 number is currently in deck. If the random numbers count in the array is smaller than 4 then a new card with the random number is created and put into the queue. Then the function calls itself again by increasing the input number of function by one.

## Card Creation

During the card creation process the card amount of the random number which is mentioned a few senteces before is increased by one. New cards number will be random number plus one as the random number will be between 0 and 12 as mentioned before. And the new cards type will be taken from cardTypes array with the current amount of number of that card in the deck, for example;

Lets say random number is 8 which means the number put into the queue will be 9

We have 2 9s already in the deck (We know this from the cardCounts array)

So new card type will be cardType[2] which is the third kind of type. { 'S', 'H', 'D', 'C' }

Then cardCount of 8 will be increased by one and the card will be put into the queue.

If the random numbers count is equal to 4 which means all of the cards for that number is already in deck, function calls itself again with the same input number as before since no card is added. This continues as long as the input parameter of function reaches to 52.

# Evaluating Process

After initializing the necessary structures and creating the deck. Program calls printDeck to write shuffled deck to output file. Then fortune telling starts from the front of the main deck queue. While front of the deck exists. Program starts counting from 1 and takes the front of the queue. If the counting number is not equal to 14 it checks if the counting number and the number of the front of the deck queue is equal, if they are equal it pops all the cards in the temporary deck and enqueues them to the main deck. Then puts the front card of the deck to the fortune deck. Then makes counting to 1 again.

If the card number and counting number is not equal. It pushes the front card to the temporary area. Then increases the counting by one.

If the counting number is equal to 14 program pops and removes all the cards in the temporary area and makes the counting number to 1 again.

# Final Calculations

After reading through all of the cards in the deck.calculateFortune function is called. This function reads through fortune queue as long as front exists and adds the number of cards to a total integer. For king queen and joker the number is limited to 10. After this, if the total is smaller then 50, this means bad luck, and otherwise user will be lucky.