CIS7 Course Project Part 2

Overview

In this course project, you will work with team members solve problems in discrete mathematics and create a C++ program that includes the representation of the solutions. This project encapsulates the application of discrete structures concepts and theories in programming.

use readme to describe - github

sumbit repo link once done

self evaluation

You will be graded on the following:

* Ability to apply concepts and theory in C++ programming: 40 points.
* Ability to provide explanation of solutions and approaches in documentation: 40 points.
* Ability to be plan, organize and work collaboratively in a team: 20 points.

Team for project:

* 2 or more members per team.
* Each team will select a project case of interest.

Project Cases:

In each team, select 2 project cases. The project scenarios are as followed:

Case 1: Inland Empire Solar Sales Travel

A solar marketing specialist travels from Riverside to Moreno Valley, then to Perris and Hemet in order to market solar packages to homeowners. The travel route is shown below:



* Assuming that the marketing specialist begins his trip from his home in Riverside and returns home daily, determine the variations of the trip to the noted cities that he must take to market the solar products.
* Determine breadth first search, shortest path, and the most low cost trips for the marketing specialist.
* Create a C++ program to provide the representation of the trips, low cost or shortest paths, including matrices, adjacencies.
* Provide documentation that explains solution and programming approaches.

Case 2: UCR Medical Center Volunteers

University of California Riverside annually selects 100 graduates to participate in Doctors without Borders program. These physicians are matched based on specialized areas, language fluency and preferences. One in every 4 participants speak his or her prefer language fluently. Each pair of participants are given opportunity to aid on-site physicians for 60 days. Currently, Doctors without Borders expand over 70 countries.

* Determine the matches of participants in the program.
* Create a C++ program that demonstrates match selection based on language and specialized areas.
* At least 7 languages must be implemented for 70 country matches.
* At least 5 medical specialization must be implemented in the matches. See <https://en.wikipedia.org/wiki/Specialty_(medicine)>
* Determine the probability of participants matching based on language fluency.
* Provide documentation that explains solution and programming approaches.

Case 3: Vigenere Cipher Decryption

Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets .The encryption of the original text is done using the Vigenère square or Vigenère table.

The table consists of the alphabets written out 26 times in different rows, each alphabet shifted ***cyclically to the left compared to the previous alphabet***, corresponding to the 26 possible Caesar Ciphers.

At different points in the encryption process, ***the cipher uses a different alphabet from one of the rows***.

The alphabet used at each point depends on a repeating keyword.

Example:

Input : Plaintext : GEEKSFORGEEKS

Keyword : AYUSH

Output : Ciphertext : GCYCZFMLYLEIM

For generating key, the given keyword is repeated in a circular manner until it matches the length of the plain text.

The keyword "AYUSH" generates the key "AYUSHAYUSHAYU"

The plain text is then encrypted using the process explained below.



*Encryption*

The first letter of the plaintext, G is paired with A, the first letter of the key. So use row G and column A of the Vigenère square, namely G. Similarly, for the second letter of the plaintext, the second letter of the key is used, the letter at row E and column Y is C. The rest of the plaintext is enciphered in a similar fashion.

*Decryption*Decryption is performed by going to the row in the table corresponding to the key, finding the position of the ciphertext letter in this row, and then using the column’s label as the plaintext. For example, in row A (from AYUSH), the ciphertext G appears in column G, which is the first plaintext letter. Next we go to row Y (from AYUSH), locate the ciphertext C which is found in column E, thus E is the second plaintext letter.

An easier **implementation** could be to visualize Vigenère algebraically by converting [A-Z] into numbers [0–25].

**Encryption**

The the plaintext(P) and key(K) are added modulo 26.

Ei = (Pi + Ki) mod 26

**Decryption**

Di = (Ei - Ki + 26) mod 26

**Note:** Di denotes the offset of the i-th character of the plaintext. Like offset of **A** is 0 and of **B** is 1 and so on.

* Given the above description of the Vigenère cipher, create a C++ program that encrypt and decrypts a message from the user.
* Test the program to verify that the output is accurate. Assess program for limitations.
* Provide documentation that explains programming approaches, algorithm is integrated in the program, and the application of course concepts.

Case 4: Casino Black Jack

Blackjack is one of the most popular casino card games in the world. The name blackjack comes from the fact that when blackjack was first introduced in the U.S. it wasn't very popular, so casinos and gambling houses tried offering different bonus payoffs. One of those was a 10-to-1 payoff for a hand consisting of the ace of spades and a black jack (that is, the jack of spades or the jack of clubs). With the current rules, a blackjack hand doesn't even need to contain a jack.

*Rules*

A blackjack game has a dealer and one or more players. Each player plays against the dealer. All players are initially dealt two cards and the dealer is dealt one card face down and one face up (these are called the hole card and up card respectively). Each player can then hit (ask for an additional card) until her total exceeds 21 (this is called busting) or she decides to stand (stop taking cards for the rest of the hand). Face cards count as 10 and an ace may be counted as 1 or 11. After all of the players have finished, the dealer reveals the hole card and plays the hand with a fixed strategy: hit on 16 or less and stand on 17 or more.

The player loses if she busts and wins if she does not bust and the dealer does (observe that if both the player and the dealer bust, the player loses). Otherwise, the player wins if her total is closer to 21 than the dealer's. If the player wins, she gets twice her bet; if she loses, she loses her money. If the dealer and player tie it is called a "push;" the player keeps her bet but does not earn any additional money. If the player's first two cards total 21, this is a blackjack and she wins 1.5 times her bet (unless the dealer also has a blackjack, in which case a tie results), so she gets back 2.5 times her bet.

Soft Hand

A hand that contains an ace that can be counted as 11 is called a soft hand, since one cannot bust by taking a card. With soft hands, the basic strategy is to always hit 17 or less and even hit 18 if the dealer's up card is 9 or 10 (where the 10 refers to a 10, J, Q, or K).

Doubling down

After the player is dealt her initial two cards she has the option of doubling her bet and asking for one additional card (which is dealt face down). The player may not hit beyond this single required card. With the basic strategy, you should always double with a total of 11, double with 10 unless the dealer's up card is 10 or A, and double with 9 only against a dealer's 2 to 6. (Some casinos only allow doubling down on 11).

Splitting pairs

At the beginning of a hand, if the player has two cards with the same number (that is, a pair) she has the option of splitting the pair and playing two hands. In principle, a pair of aces should of course be split, but in this case blackjack rules allow you to get only one card on each hand, and getting a 10 does not make a blackjack. With the basic strategy, you should never split 10's, 5's or 4's, always split 8's, and, in the other cases, split against an up card of 2 to 7, but not otherwise.

Strategies for the Player

Blackjack is almost always disadvantageous for the player, meaning that no strategy yields a positive expected payoff for the player. In the long run, whatever you do, you will on average lose money. Exceptions exist: some casinos offer special rules that allow a player using the right strategy to have a positive expected payoff; such casinos are counting on the players making mistakes.

This basic strategy is based on the player's point total and the dealer's visible card. It consists of a table that describes what you should do in any situation in the game.



Key:

**S** = Stand

**H** = Hit

**Dh** = Double (if not allowed, then hit)

**Ds** = Double (if not allowed, then stand)

**SP** = Split

**SU** = Surrender (if not allowed, then hit)

Under the most favorable set of rules, the house advantage against a player using the basic strategy can be as low as 0.16%.



Many people assume that the best strategy for the player is to mimic the dealer. A second conservative strategy is called never bust: hit 11 or less, stand on 12 or more. Each of these strategies leads to a player disadvantage of about 6%.

* Based on the above description of Black Jack, create a C++ program that represents the Black Jack casino game using **single card deck (52 cards)**. In the program, **provide player advantage probability in for each hand**.
* The program should show dealer hand and player hand.
* The winning probability for each player hand before and after card hit.
* In the documentation, include your programming approaches and solutions that apply course concepts.