```
#include <iostream>
#include <random>
#include <ctime>

// Function to generate a random number within a given range
int getRandomNumber(int min, int max) {
        //compiling psuedo random pattern generated values in an
   //initialized range using a "PRBS standard"
        static std::mt19937 rand(static_cast<unsigned int>(time(0)));
   return std::uniform_int_distribution<>(min, max)(rand);
}

// Function to display the cards and total value of a user
void displayUserCards(const std::string& username, const std::string* cards, const std::string* suits, int numCards, int total) {
   std::cout << username << " has the following cards:" << std::endl;
   for (int i = 0; i < numCards; ++i) {
      std::cout << cards[i] << " of " << suits[i] << std::endl;
   }
   std::cout << "Total value: " << total << std::endl;
}

// Function to get the value of a card
int getCardValue(const std::string& card) {
   if (card == "Ace")
      return 11;
   if (card == "King" || card == "Queen" || card == "Jack")
      return 10;
   return std::stoi(card);
}

int main() {
```

```
        // Seed the random number generator
srand(static_cast<unsigned int>(time(0)));

const int numCards = 13;
const int numSuits = 4;
const std::string cards[numCards] = {"Ace", "2", "3", "4", "5", "6", "7", "8", "9", "10", "King", "Queen", "Jack"};
const std::string suits[numSuits] = {"Hearts", "Diamonds", "Clubs", "Spades"};

std::string user1Cards[4];
std::string user1Suits[4];
int user1Total = 0;

std::string user2Cards[4];
std::string user2Suits[4];
int user2Total = 0;

// User 1 receives two initial cards
for (int i = 0; i < 2; ++i) {
   int cardIndex = getRandomNumber(0, numCards - 1);
   int suitIndex = getRandomNumber(0, numSuits - 1);
   user1Cards[i] = cards[cardIndex];
   user1Suits[i] = suits[suitIndex];
   user1Total += getCardValue(user1Cards[i]);

   // Display the card received by user 1
   std::cout << "User 1 received: " << user1Cards[i] << " of " << user1Suits[i] << std::endl;
}

// Display the total value of user 1 with the initial cards
displayUserCards("User 1", user1Cards, user1Suits, 2, user1Total);
```

```
// Calculate the probability of user 1 exceeding 21 with the initial cards
int favorableCards = 0;
int possibleCards = numCards * numSuits;
for (int i = 0; i < numCards; ++i) {
    for (int j = 0; j < numSuits; ++j) {
        int value = getCardValue(cards[i]);
        if (user1Total + value > 21)
            ++favorableCards;
    }
}

double probability = static_cast<double>(favorableCards) / possibleCards * 100;
std::cout << "Probability of User 1 exceeding 21 with the initial cards: " << probability << "%" << std::endl;

// Check if user 1 wins immediately with an ace and a face card
if ((user1Cards[0] == "Ace" && (user1Cards[1] == "King" || user1Cards[1] == "Queen" || user1Cards[1] == "Jack")) ||
    (user1Cards[1] == "Ace" && (user1Cards[0] == "King" || user1Cards[0] == "Queen" || user1Cards[0] == "Jack"))) {
    std::cout << "User 1 wins!" << std::endl;
    return 0;
}

// User 1 chooses to receive additional cards
char choice;
do {
    std::cout << "Do you want to receive an additional card? (Y/N): ";
    std::cin >> choice;

    if (choice == 'Y' || choice == 'y') {
        int cardIndex = getRandomNumber(0, numCards - 1);
        int suitIndex = getRandomNumber(0, numSuits - 1);
        user1Cards[2] = cards[cardIndex];
        user1Suits[2] = suits[suitIndex];
```

```
        int cardValue = getCardValue(user1Cards[2]);
        user1Total += cardValue;

        // Display the additional card received by user 1
        std::cout << "User 1 received an additional card: " << user1Cards[2] << " of " << user1Suits[2] << std::endl;

        // Display the new total value of user 1
        displayUserCards("User 1", user1Cards, user1Suits, 3, user1Total);

        // Calculate the probability of user 1 exceeding 21 with the additional card
        if (user1Total <= 21) {
            ++favorableCards;
            probability = static_cast<double>(favorableCards) / possibleCards * 100;
            std::cout << "Probability of User 1 exceeding 21 with an additional card: " << abs(1 - probability) << "%" << std::endl;
        }

        // Check if user 1 exceeds 21 and loses
        if (user1Total > 21) {
            std::cout << "User 1 loses!" << std::endl;
            return 0;
        }
    }
  } while (choice == 'Y' || choice == 'y');

  // User 2 receives two initial cards
  for (int i = 0; i < 2; ++i) {
    int cardIndex = getRandomNumber(0, numCards - 1);
    int suitIndex = getRandomNumber(0, numSuits - 1);
    user2Cards[i] = cards[cardIndex];
    user2Suits[i] = suits[suitIndex];
    user2Total += getCardValue(user2Cards[i]);
```

```
    // Display the card received by user 2
    std::cout << "User 2 received: " << user2Cards[i] << " of " << user2Suits[i] << std::endl;
}

// Display the total value of user 2 with the initial cards
displayUserCards("User 2", user2Cards, user2Suits, 2, user2Total);

// Check if user 2 wins immediately with an ace and a face card
if ((user2Cards[0] == "Ace" && (user2Cards[1] == "King" || user2Cards[1] == "Queen" || user2Cards[1] == "Jack")) ||
    (user2Cards[1] == "Ace" && (user2Cards[0] == "King" || user2Cards[0] == "Queen" || user2Cards[0] == "Jack"))) {
    std::cout << "User 2 wins!" << std::endl;
    return 0;
}

// Check if user 1 has a total value of 21 and wait for user 2
if (user1Total == 21) {
    std::cout << "User 1 has a total value of 21. Waiting for User 2..." << std::endl;
    while (user2Total < 21) {
        int cardIndex = getRandomNumber(0, numCards - 1);
        int suitIndex = getRandomNumber(0, numSuits - 1);
        user2Cards[2] = cards[cardIndex];
        user2Suits[2] = suits[suitIndex];
        int cardValue = getCardValue(user2Cards[2]);
        user2Total += cardValue;

        // Display the additional card received by user 2
        std::cout << "User 2 received an additional card: " << user2Cards[2] << " of " << user2Suits[2] << std::endl;

        // Display the new total value of user 2
        displayUserCards("User 2", user2Cards, user2Suits, 3, user2Total);
```

```
        // Check if user 2 exceeds 21 and loses
        if (user2Total > 21) {
            std::cout << "User 2 loses. User 1 wins!" << std::endl;
            return 0;
        }
    }
    // Check if user 2 has a total value of 21
    if (user2Total == 21) {
        std::cout << "There are no losers!" << std::endl;
    } else {
        std::cout << "User 2 wins!" << std::endl;
    }
} else {
    // User 2 continues to receive cards until they exceed 21 or have a greater value than user 1
    while (user2Total <= user1Total) {
        int cardIndex = getRandomNumber(0, numCards - 1);
        int suitIndex = getRandomNumber(0, numSuits - 1);
        user2Cards[2] = cards[cardIndex];
        user2Suits[2] = suits[suitIndex];
        int cardValue = getCardValue(user2Cards[2]);
        user2Total += cardValue;

        // Display the additional card received by user 2
        std::cout << "User 2 received an additional card: " << user2Cards[2] << " of " << user2Suits[2] << std::endl;

        // Display the new total value of user 2
        displayUserCards("User 2", user2Cards, user2Suits, 3, user2Total);

        // Check if user 2 exceeds 21 and loses
        if (user2Total > 21) {
            std::cout << "User 2 loses. User 1 wins!" << std::endl;
            return 0;
```

```
            }
        }
        std::cout << "User 2 wins!" << std::endl;
    }

    return 0;
}
```