```cpp
// DAAL
#include <iostream>
#include <cstdlib>
#include <ctime>
#include <vector>
#include <string>
#include <sstream>
using namespace std;
string dealCard();
bool checkDuplicate(string hand[], string aCard);
void dealHand(string hand[]);
void swapCards(string hand[]);
void convertToInt(string hand[], int numHand[2][5]);
string findWinner(string hand[]);
void game();

char board[3][3] = {{'1', '2', '3'}, {'4', '5', '6'}, {'7', '8', '9'}};
char current_marker;
int current_player;

void reset()
{
    board[0][0] = '1';
    board[0][1] = '2';
    board[0][2] = '3';
    board[1][0] = '4';
    board[1][1] = '5';
    board[1][2] = '6';
    board[2][0] = '7';
    board[2][1] = '8';
    board[2][2] = '9';
}
int main()
{
    srand(time(NULL));
    int option, choice;
    bool stop = false;

    while (stop != true)
    {
        cout << "****GROUP 6 GAME****" << endl;
        cout << "1.Card game" << endl;
        cout << "2.Tic Tac Toe" << endl;
        cout << "3.Exit" << endl
             << endl;
        cout << "Choose your option : ";
        cin >> option;
        switch (option)
        {
        case 1:
```

```
51:             if (option == 1)
52:             {
53:                 string hand[5] = {"", "", "", "", ""};
54:                 dealHand(hand);
55:                 for (int i = 0; i < 5; i++)
56:                     cout << endl
57:                          << (i + 1) << " - " << hand[i];
58:
59:                 // Give the player the option to swap their cards
60:                 char yesOrNo;
61:                 cout << endl
62:                      << endl
63:                      << "Do you want to change cards? (y/n): ";
64:                 cin >> yesOrNo;
65:                 if (yesOrNo == 'y')
66:                 {
67:                     swapCards(hand);
68:                     cout << endl
69:                          << "Your new cards are:" << endl;
70:                     for (int i = 0; i < 5; i++)
71:                         cout << (i + 1) << " - " << hand[i] << endl;
72:                     cout << endl;
73:                 }
74:
75:                 // Find a winner in the hand
76:                 cout << endl
77:                      << findWinner(hand) << endl;
78:
79:                 // Pause program
80:                 cin.ignore(256, '\n');
81:             }
82:             break;
83:         case 2:
84:             game();
85:             break;
86:         case 3:
87:             cout << "------Thank you!!!------";
88:             return 0;
89:         default:
90:             cout << "\n!!!You selected an invalid choice!!!\n";
91:             break;
92:         }
93:         cout<<"*\n";
94:         cout<<"Would you like to play again :\n";
95:         cout<<"PRESS 1 for Yes\n";
96:         cout<<"PRESS 0 for No : ";
97:         cin>>choice;
98:
99:         system("CLS"); // CLS is for clearing the system
100:        if (choice == 0)
```

```
101:         {
102:             cout<<"**Thank You**";
103:             return 0;
104:         }
105:     }
106:     return 0;
107: }
108: string dealCard()
109: {
110:     string card;
111:
112:     int aSuit = (rand() % (4 - 1 + 1)) + 1;
113:     int aCard = (rand() % (13 - 1 + 1)) + 1;
114:
115:     string face;
116:     switch (aSuit)
117:     {
118:     case 1:
119:         face = " of Hearts";
120:         break;
121:     case 2:
122:         face = " of Spades";
123:         break;
124:     case 3:
125:         face = " of Diamonds";
126:         break;
127:     case 4:
128:         face = " of Clubs";
129:         break;
130:     }
131:     string value;
132:     switch (aCard)
133:     {
134:     case 1:
135:         value = "Ace";
136:         break;
137:     case 2:
138:         value = "2";
139:         break;
140:     case 3:
141:         value = "3";
142:         break;
143:     case 4:
144:         value = "4";
145:         break;
146:     case 5:
147:         value = "5";
148:         break;
149:     case 6:
150:         value = "6";
```

```
151:         break;
152:     case 7:
153:         value = "7";
154:         break;
155:     case 8:
156:         value = "8";
157:         break;
158:     case 9:
159:         value = "9";
160:         break;
161:     case 10:
162:         value = "10";
163:         break;
164:     case 11:
165:         value = "Jack";
166:         break;
167:     case 12:
168:         value = "Queen";
169:         break;
170:     case 13:
171:         value = "King";
172:         break;
173:     }
174:     card = (value + face);
175:     return card;
176: }
177: bool checkDuplicate(string hand[], string aCard)
178: {
179:     bool duplicate = false;
180:
181:     for (int i = 0; i < 5; i++)
182:     {
183:         if (hand[i] == aCard)
184:             duplicate = true;
185:     }
186:
187:     return duplicate;
188: }
189: void dealHand(string hand[])
190: {
191:     for (int i = 0; i < 5; i++)
192:     {
193:         string j = dealCard();
194:         if (checkDuplicate(hand, j) == true)
195:             i--;
196:         else
197:             hand[i] = j;
198:     }
199: }
200: void swapCards(string hand[])
```

```
{
    char yesOrNo;

    for (int i = 0; i < 5; i++)
    {
        cout << "Change card " << (i + 1) << "? (y/n): ";
        cin.ignore(256, '\n');
        cin >> yesOrNo;

        if (yesOrNo == 'y')
        {
            string j;
            do
            {
                j = dealCard();
            } while (checkDuplicate(hand, j) == true);
            hand[i] = j;
        }
        else
        {
            continue;
        }
    }
}
void convertToInt(string hand[], int numHand[2][5])
{
    stringstream stream;

    for (int i = 0; i < 5; i++)
    {
        if (hand[i].substr(0, 2) == "10")
        {
            numHand[0][i] = 10;
        }
        else if (hand[i].substr(0, 1) == "A")
        {
            numHand[0][i] = 1;
        }
        else if (hand[i].substr(0, 1) == "J")
        {
            numHand[0][i] = 11;
        }
        else if (hand[i].substr(0, 1) == "Q")
        {
            numHand[0][i] = 12;
        }
        else if (hand[i].substr(0, 1) == "K")
        {
            numHand[0][i] = 13;
        }
```

```
251:         else
252:         {
253:             stream << hand[i].substr(0, 1);
254:             stream >> numHand[0][i];
255:             stream.str("");
256:             stream.clear();
257:         }
258:     }
259: 
260:     enum
261:     {
262:         HEARTS,
263:         SPADES,
264:         DIAMONDS,
265:         CLUBS
266:     };
267: 
268:     for (int i = 0; i < 5; i++)
269:     {
270:         if (hand[i].find("Hearts") != -1)
271:         {
272:             numHand[1][i] = HEARTS;
273:         }
274:         else if (hand[i].find("Spades") != -1)
275:         {
276:             numHand[1][i] = SPADES;
277:         }
278:         else if (hand[i].find("Diamonds") != -1)
279:         {
280:             numHand[1][i] = DIAMONDS;
281:         }
282:         else
283:         {
284:             numHand[1][i] = CLUBS;
285:         }
286:     }
287: }
288: // Evaluate whether there is a winner in the hand
289: string findWinner(string hand[])
290: {
291:     string result = "Sorry, better luck next time!";
292: 
293:     bool straightFlush = false;
294:     bool fourOfAKind = false;
295:     bool fullHouse = false;
296:     bool flush = false;
297:     bool straight = false;
298:     bool threeOfAKind = false;
299:     bool twoPair = false;
300:     bool onePair = false;
```

```
301:
302:     enum
303:     {
304:         PAIR = 1,
305:         TWO_PAIR,
306:         THREE_OF_A_KIND,
307:         FULL_HOUSE,
308:         FOUR_OF_A_KIND = 6
309:     };
310:
311:     int cards[2][5];
312:     convertToInt(hand, cards);
313:     // Looks for a pair, two pair, three of a kind,
314:     // full house and four of a kind.
315:     vector<int> winners(0);
316:     for (int i = 0; i < 4; i++)
317:     {
318:         for (int j = i; j < 4; j++)
319:         {
320:             if (cards[0][i] == cards[0][j + 1])
321:                 winners.push_back(cards[0][i]);
322:         }
323:     }
324:     if (winners.size() == FOUR_OF_A_KIND)
325:         fourOfAKind = true;
326:     else if (winners.size() == FULL_HOUSE)
327:         fullHouse = true;
328:     else if (winners.size() == THREE_OF_A_KIND)
329:         threeOfAKind = true;
330:     else if (winners.size() == TWO_PAIR)
331:         twoPair = true;
332:     else if (winners.size() == PAIR)
333:         onePair = true;
334:
335:     // Looks for a straight
336:     vector<int> orderCards(0);
337:     for (int i = 0; i < 5; i++)
338:         orderCards.push_back(cards[0][i]);
339:     // Bubble sort algorithm
340:     bool swapped = true;
341:     int j = 0;
342:     int tmp;
343:     while (swapped)
344:     {
345:         swapped = false;
346:         j++;
347:         for (int i = 0; i < (orderCards.size() - j); i++)
348:         {
349:             if (orderCards[i] > orderCards[i + 1])
350:             {
```

```
351:                 tmp = orderCards[i];
352:                 orderCards[i] = orderCards[i + 1];
353:                 orderCards[i + 1] = tmp;
354:                 swapped = true;
355:             }
356:         }
357:     }
358:     int checkStraight = orderCards.back();
359:     while (checkStraight != orderCards.front())
360:     {
361:         checkStraight--;
362:     }
363:     if (checkStraight == (orderCards.back() - (orderCards.size() - 1)))
364:     {
365:         straight = true;
366:     }
367:     else if ((orderCards[0] == 1) &&
368:              (orderCards[1] == 10) &&
369:              (orderCards[2] == 11) &&
370:              (orderCards[3] == 12) &&
371:              (orderCards[4] == 13))
372:     {
373:         straight = true;
374:     }
375:     // Looks for a flush
376:     for (int i = 0; i < 4; i++)
377:     {
378:         if (cards[1][i] == cards[1][i + 1])
379:         {
380:             flush = true;
381:         }
382:         else
383:         {
384:             flush = false;
385:             break;
386:         }
387:     }
388:     // Looks for a straight flush
389:     if ((straight == true) && (flush == true))
390:         straightFlush = true;
391:
392:     if (straightFlush == true)
393:         result = "You have a Straight Flush!"; // a poker hand containing five cards
394:     else if (fourOfAKind == true)
395:         result = "You have a Four Of A Kind!"; // a hand that contains four cards of
396:     else if (fullHouse == true)
397:         result = "You have a Full House!"; // a hand that contains three cards of or
398:     else if (flush == true)
399:         result = "You have a Flush!"; // a hand that contains five cards all of the
400:     else if (straight == true)
```

```
401:         result = "You have a Straight!"; // a hand that contains five cards of seque
402:     else if (threeOfAKind == true)
403:         result = "You have a Three Of A Kind!"; // a five-card hand in which three c
404:     else if (twoPair == true)
405:         result = "You have a Two Pair!"; // hand that contains 2 cards of one rank,2
406:     else if (onePair == true)
407:         result = "You have a Pair!"; // a poker hand where we have two cards of iden
408: 
409:     return result;
410: }
411: 
412: void drawBoard()
413: {
414:     cout << "_____________\n";
415:     cout << "| " << board[0][0] << " | " << board[0][1] << " | " << board[0][2] << "
416:     cout << "_____________\n";
417:     cout << "| " << board[1][0] << " | " << board[1][1] << " | " << board[1][2] << "
418:     cout << "_____________\n";
419:     cout << "| " << board[2][0] << " | " << board[2][1] << " | " << board[2][2] << "
420:     cout << "_____________\n";
421: }
422: 
423: bool placeMarker(int slot)
424: {
425:     int row = slot / 3;
426:     int col;
427: 
428:     if (slot % 3 == 0)
429:     {
430:         row = row - 1;
431:         col = 2;
432:     }
433:     else
434:     {
435:         col = slot % 3 - 1;
436:     }
437:     if (board[row][col] != 'X' && board[row][col] != 'O')
438:     {
439:         board[row][col] = current_marker;
440:         return true;
441:     }
442:     else
443:     {
444:         return false;
445:     }
446: }
447: 
448: int winner()
449: {
450:     for (int i = 0; i < 3; i++)
```

```
451:     {
452:         // rows
453:         if (board[i][0] == board[i][1] && board[i][1] == board[i][2])
454:         {
455:             return current_player;
456:         }
457:         // columns
458:         if (board[0][i] == board[1][i] && board[1][i] == board[2][i])
459:         {
460:             return current_player;
461:         }
462:     }
463:     if (board[0][0] == board[1][1] && board[1][1] == board[2][2])
464:     {
465:         return current_player;
466:     }
467:     if (board[0][2] == board[1][1] && board[1][1] == board[2][0])
468:     {
469:         return current_player;
470:     }
471:     return 0;
472: }
473:
474: void swap_player_and_marker()
475: {
476:     if (current_marker == 'X')
477:     {
478:         current_marker = 'O';
479:     }
480:     else
481:     {
482:         current_marker = 'X';
483:     }
484:     if (current_player == 1)
485:     {
486:         current_player = 2;
487:     }
488:     else
489:     {
490:         current_player = 1;
491:     }
492: }
493:
494: void game()
495: {
496:     cout << "Player one, choose your marker from 'O' or 'X' :";
497:     char marker_p1;
498:     cin >> marker_p1;
499:
500:     current_player = 1;
```

```
501:     current_marker = marker_p1;
502:     reset();
503:     drawBoard();
504:
505:     int player_won;
506:
507:     for (int i = 0; i < 9; i++)
508:     {
509:         cout << "It`s player " << current_player << "`s turn. Enter your slot: ";
510:         int slot;
511:         cin >> slot;
512:
513:         if (slot < 1 || slot > 9)
514:         {
515:             cout << "That slot is invalid! Try another slot!";
516:             i--;
517:             continue;
518:         }
519:
520:         if (!placeMarker(slot))
521:         {
522:             cout << "That slot occupied! Try another slot!";
523:             i--;
524:             continue;
525:         }
526:         drawBoard();
527:
528:         player_won = winner();
529:
530:         if (player_won == 1)
531:         {
532:             cout << "Player 1 won! Congratulations!";
533:             // reset();
534:             break;
535:         }
536:         if (player_won == 2)
537:         {
538:             cout << "Player 2 won! Congratulations!";
539:             // reset();
540:             break;
541:         }
542:
543:         swap_player_and_marker();
544:     }
545:     if (player_won == 0)
546:     {
547:         cout << "That is a Tie!";
548:         // reset();
549:     }
550: }
```