

Blackjack.cpp

```
1 //=====
2 // Name      : Blackjack.cpp
3 // Author    : douglas holmes
4 // Version   : 1.0
5 // Copyright : ©holmes2018
6 // Description : Blackjack
7 //=====
8 /*****
9 Blackjack C++ routine created by Douglas Holmes 11-2018 to 12-2018 ©holmes2018.
10 This game is based on the instructions for blackjack on the "Bicycle cards" web site.
11 One player, plays the computer dealer, and can split up to four times. Chips are
12 offered at the beginning of the game and can be added during play. Hit, stay, split,
13 double down, and print-hand are commands the player can choose. Six new ordered decks
14 of cards are opened and shuffled four times. The player is then asked to cut them
15 before they are placed in a shoe (stack). This program was created using the Eclipse
16 compiler on a Mac. I created this program as a final project for a MIT Open Courseware
17 class #6-096.
18 *****/
19 /
20
21 #include <iostream>
22 #include <stdio.h>
23 #include <stdlib.h>
24 #include <time.h>
25 #include <unistd.h>
26 // #include <wchar>
27 // #include <string>
28 // #include <wchar.h>
29 using namespace std;
30 // global variable
31 int totalplayers = 1; // player
32 int splittime[6] = {0}; // [0] is used to set the hand that is going to be split
33 int cardT;
34 int bump;
35
36 // declare prototypes
37 // a struct to create "shoe" of cards
38 typedef struct { int *stk ; int maxelem; int top; } STACK;
39 STACK *init(int n);
40 void fillStack(STACK *s, int arr[], int x);
41 void push(STACK *s, int data);
42 int pop(STACK *s);
43 void deleteNode(STACK *s, int data);
44 void hit(STACK *s, int pno, int a[], int count[pno]);
45 void dealCards(STACK *s, int player[], int house[], int count[]);
46 void CheckSplit (STACK *s, int a[], int b[], int count[totalplayers], double bet[], int
splitplayer);
47 string cardName(int a[]);
48 string cardNamed(int a);
49 int cardValue(int a[]);
```

Blackjack.cpp

```

50 int winner(int a[], int bank, double bet[], int natural[]);
51 void newDeck (int x, int arr[x]);
52 void printArray (int cardArr[], string whoIs, int pno, double bet[pno], int n);
53 void swap (int *a, int *b);
54 void randomize ( int arr[], int n);
55 void cutCards(int arr[], int x, int c);
56 int checkNatural(int a[], int b[], double bet[], int pno);
57 int cardDealer(int a);
58 void doubleDown(STACK *s, int pno, int a[], int count[], double bet[]);
59
60
61 // Driver program to test above function.
62 int main()
63 {
64     string hand[6]={"Dealer  ", "Original ", "Split 1 ", "Split 2 ", "Split 3 ",
"Split 4 "};
65     int x = 52 * 6;
66     cardT = x;
67     int w = 0;
68     int c;
69     double bet[6]={0};
70     int bank = 20;
71     int reshuffling;
72     char resp;
73     int *p;
74     int *p1;
75     int *cc;
76     string CreditCard;
77     STACK *s = NULL;
78     int arr[x];
79     s=init(x);
80     int n = sizeof(arr)/ sizeof(arr[0]);
81
82     cout <<"Blackjack based on Bicycle playing car rules (6 decks of cards).....
\n\n";
83     newDeck (x, arr);
84     randomize (arr, n);
85     randomize (arr, n);
86     randomize (arr, n);
87     randomize (arr, n);
88     cout << "Cut cards, pick a number between 1 and " << x << ": \n";
89     while(!(cin >> c))
90     {
91         cout<<"Invalid data type! Please enter 'a number' between 1 and " << x;
92         cin.clear();
93         cin.ignore(999, '\n');
94     }
95     if(c < 1 || c > x){c = 1;}
96
97     cutCards(arr, x, c);
98     fillStack(s, arr, x);

```

Blackjack.cpp

```

99  srand (time(NULL)); /* initialize random seed: */
100 reshuffling = rand() % 25 + 26; /* generate a number between 25 and 50: */
101 cout << "The Card Shoe is filled and the reshuffling card has been placed.\n\n";
102 cout << "Chips have the value of $10.00, How many chips would you like to Buy? \n";
103 while(!(cin >> bank))
104     {
105         cout<<"Invalid data type! Please enter 'a number'";
106         cin.clear();
107         cin.ignore(999, '\n');
108     }
109
110 cout << "Please put in your Credit Card Number:";
111 while(!(cin >> CreditCard))
112     {
113         cout<<"Invalid data type! Please enter 'a number'";
114         cin.clear();
115         cin.ignore(999, '\n');
116     }
117
118
119 cout << "processing order \n";
120 int j = 0;
121 while(j < 100){cout << "."; usleep(10000); j++;}
122 cout << "\n";
123 cout << "That is not a valid credit card, leave your watch and I'll let you play
124 anyway.....\n\n";
125 cout << "You have " << bank << " chips, Please place your bet:";
126 while(!(cin >> bet[1]))
127     {
128         cout<<"Invalid data type! Please enter 'an even number' between 1 and
129 " << bank;
130         cin.clear();
131         cin.ignore(999, '\n');
132     }
133 while(bet[1] != 0)
134 {
135     int insurance[3] = {0};
136     int natural[6] = {0};
137     int dealerhit = 0;
138     int count[6]= {0};
139     int scores[6]={0};
140     int house[10] = {0};
141     int player[10] = {0};
142     int split1[10] = {0};
143     int split2[10] = {0};
144     int split3[10] = {0};
145     int split4[10] = {0};
146     totalplayers = 1;
147     splittime[0] = 1;
148     splittime[1] = 1;

```

Blackjack.cpp

```

148     int currentplayer = 1;
149     w = 1;
150     p = player;
151     p1 = house;
152     cc = count;
153     dealCards(s, p, p1, count);
154     printArray (p1, hand[0] ,0, bet, 0);
155     // force Split/natural
156     //   p[0] = 1;
157     //   p[1] = 14;
158     printArray (p, hand[currentplayer],currentplayer , bet, w);
159     insurance[0] = cardDealer(house[0]);
160     insurance[1] = cardDealer(house[1]);
161     if(insurance[0] == 11)
162     {
163         cout << "Would you like to buy insurance (input a value between 0 and " << bank -
bet[1] << " )?\n";
164         while(!(cin >> insurance[2]))
165             {
166                 cout<<"Invalid data type! Please enter 'a number'";
167                 cin.clear();
168                 cin.ignore(999, '\n');
169             }
170         bank -= insurance[2];
171         if(insurance[0] + insurance[1] == 21 && insurance[2] > 0 )
172         {
173             bank += insurance[2] * 2;
174             cout << "Good thing you had insurance, the dealer has 21, your bank is now " <<
bank << " chips\n";
175             currentplayer = totalplayers + 1;
176         }else if(insurance[0] + insurance[1] == 21 && insurance[2] == 0 )
177         {
178             bank -= insurance[2] + bet[1];
179             cout << "Should have had some insurance dealer has 21, your bank is now "
<< bank << " chips\n";
180             currentplayer = totalplayers + 1;
181         }else if(insurance[0] + insurance[1] != 21 && insurance[2] > 0 )
182         {
183             bank -= insurance[2];
184             cout << "Dealer does not have 21 you loose your side bet, your bank is
now " << bank << " chips\n";
185         }else
186         {cout << "dealer does not have 21 \n";}
187         }
188     splittime[0] = 1;
189     while(currentplayer <= totalplayers)
190     {
191         if(currentplayer == 1){p = player;}
192         if(currentplayer == 2){p = split1;}
193         if(currentplayer == 3){p = split2;}
194         if(currentplayer == 4){p = split3;}

```

Blackjack.cpp

```

195     if(currentplayer == 5){p = split4;}
196     if(totalplayers == 1){p1 = split1;}
197     if(totalplayers == 2){p1 = split2;}
198     if(totalplayers == 3){p1 = split3;}
199     if(totalplayers == 4){p1 = split4;}
200
201
202     if(currentplayer > 1){hit(s, currentplayer, p , cc);}
203     if(currentplayer > 1){cout << ".....Playing hand " <<
hand[currentplayer] << " .....\\n"; }
204     if(currentplayer > 1){printArray (p, hand[currentplayer],currentplayer , bet, w);}
205     if(currentplayer == 1){natural[currentplayer] = checkNatural (p, house, bet,
currentplayer);}
206     if(natural[1] < 0){bank -= natural[currentplayer]; cout << "Your total is now "
<< bank << " chips \\n"; currentplayer = totalplayers + 1;}
207     if(natural[1] > 1){bank += natural[currentplayer]; cout << "Your total is now "
<< bank << " chips \\n"; currentplayer = totalplayers + 1;}
208     if(natural[1] > 1){currentplayer = currentplayer + 1; break;}
209     if(currentplayer > 1){natural[currentplayer] = checkNatural (p, house, bet,
currentplayer);}
210     while(splittime[0] == 4)
211     {
212         if(totalplayers == 1){p1 = split1;}
213         if(totalplayers == 2){p1 = split2;}
214         if(totalplayers == 3){p1 = split3;}
215         if(totalplayers == 4){p1 = split4;}
216         CheckSplit(s, p, p1, cc, bet, 4);
217         if(splittime[0] == 4 && totalplayers == 5){cout << "you have slit your
hand.....\\n";}
218         if(splittime[0] == 4 && totalplayers == 5){printArray (split3, "Split 3 ",
4, bet, w);}
219         if(splittime[0] == 4 && totalplayers == 5){printArray (split4, "Split 4 ",
5, bet, w);}
220         cout <<
".....
.\\n";
221     }
222
223     while(splittime[0] == 3)
224     {
225         if(totalplayers == 1){p1 = split1;}
226         if(totalplayers == 2){p1 = split2;}
227         if(totalplayers == 3){p1 = split3;}
228         if(totalplayers == 4){p1 = split4;}
229         CheckSplit(s, p, p1, cc, bet, 3);
230         if(splittime[0] == 3 && totalplayers == 4){cout << "you have slit your
hand.....\\n";}
231         if(splittime[0] == 3 && totalplayers == 4){printArray (split2, "Split 2 ",
3, bet, w);}
232         if(splittime[0] == 3 && totalplayers == 4){printArray (split3, "Split 3 ",
4, bet, w);}

```

Blackjack.cpp

```

233         if(splittime[0] == 3 && totalplayers == 5){printArray (split4, "Split 4 ",
5, bet, w);}
234         cout <<
".....
.\n";
235     }
236
237     while(splittime[0] == 2)
238     {
239         if(totalplayers == 1){p1 = split1;}
240         if(totalplayers == 2){p1 = split2;}
241         if(totalplayers == 3){p1 = split3;}
242         if(totalplayers == 4){p1 = split4;}
243         CheckSplit(s, p, p1, cc, bet, 2);
244         if(splittime[0] == 2 && totalplayers == 3 ){cout << "you have slit your
hand.....\n";}
245         if(splittime[0] == 2 && totalplayers == 3 ){printArray (split1, "Split 1 ",
2, bet, w);}
246         if(splittime[0] == 2 && totalplayers == 3 ){printArray (split2, "Split 2 ",
3, bet, w);}
247         if(splittime[0] == 2 && totalplayers == 4 ){printArray (split3, "Split 3 ",
4, bet, w);}
248         if(splittime[0] == 2 && totalplayers == 5 ){printArray (split4, "Split 4 ",
5, bet, w);}
249         cout <<
".....
.\n";
250     }
251
252     while(splittime[0] == 1)
253     {
254         if(totalplayers == 1){p1 = split1;}
255         if(totalplayers == 2){p1 = split2;}
256         if(totalplayers == 3){p1 = split3;}
257         if(totalplayers == 4){p1 = split4;}
258         CheckSplit(s, p, p1, cc, bet, 1);
259
260         if(splittime[0] == 1 && totalplayers == 2 ){cout << "you have slit your
hand.....\n";}
261         if(splittime[0] == 1 && totalplayers == 2 ){printArray (player, "Original ",
1 , bet, w);}
262         if(splittime[0] == 1 && totalplayers == 2){printArray (split1, "Split 1 ",
2, bet, w);}
263         if(splittime[0] == 1 && totalplayers == 3){printArray (split2, "Split 2 ",
3, bet, w);}
264         if(splittime[0] == 1 && totalplayers == 4){printArray (split3, "Split 3 ",
4, bet, w);}
265         if(splittime[0] == 1 && totalplayers == 5){printArray (split4, "Split 4 ",
5, bet, w);}
266         cout <<
".....

```

Blackjack.cpp

```

.\n";
267     }
268
269     int now = currentplayer;
270     while(now == currentplayer)
271     {
272         if(cardValue(p) >= 21){resp = 's';
273         }else if(cardValue(p) < 21){cout << "What would you like to do for your "<<
hand[currentplayer] << " hand (? for help)?\n" ;}
274         while(!(cin >> resp))
275         {
276             cout <<"Invalid data type! Please enter 'y' or 'n'";
277             cin.clear();
278             cin.ignore(999, '\n');
279         }
280
281         if (resp == 'h' || resp == 'H'){
282             hit(s, currentplayer, p , cc);
283             printArray (p, hand[currentplayer],currentplayer , bet, w);
284         } else if (resp == 'd' || resp == 'D'){
285             cout << "Doubling
down.....\n";
286             doubleDown(s, currentplayer, p, cc, bet);
287             printArray (p, hand[currentplayer],currentplayer , bet, w);
288             now -= now;
289         } else if (resp == 's' || resp == 'S')
290             { now -= now; cout <<
"Staying.....\n";
291         } else if (resp == 'p' || resp == 'P')
292             { printArray (p, hand[currentplayer],currentplayer , bet, w);
293         } else if (resp == 'b' || resp == 'B')
294             { cout << "Chips have the value of $10.00, How many chips would you
like to Buy? \n";
295             while(!(cin >> bank))
296             {
297                 cout<<"Invalid data type! Please enter 'a number'";
298                 cin.clear();
299                 cin.ignore(999, '\n');
300             }
301         } else if (resp == '?' || resp == '/')
302             {cout << "key commands you can use. h = hit, s = stay,
d = double down, p = print hand, b = buy chips ? = Help  \n";
303
304             }else {cout << "bubba \n";}
305         if(cardValue(p) >= 21){now -= now;}
306
307     }
308     currentplayer++;
309 }
310
311 if(split1[0] !=0 ){ printArray (split1, "Split 1 ", 2, bet, w);}

```

Blackjack.cpp

```

312  if(split2[0] !=0){ printArray (split2, "Split 2 ", 3, bet, w);}
313  if(split3[0] !=0){ printArray (split3, "Split 3 ", 4, bet, w);}
314  if(split4[0] !=0){ printArray (split4, "Split 4 ", 5, bet, w);}
315
316  int checkhit = 1;
317  if(natural[1] > 1 && currentplayer >= 2){checkhit = 0;}
318  while(checkhit !=0)
319  {
320      cout << "The house is thinking about hitting or
not.....\n";
321      scores[1] = cardValue(player);
322      if(split1[0] != 0 ){ scores[2] = cardValue(split1);}
323      if(split2[0] != 0 ){ scores[3] = cardValue(split1);}
324      if(split3[0] != 0 ){ scores[4] = cardValue(split1);}
325      if(split4[0] != 0 ){ scores[5] = cardValue(split1);}
326
327
328      if(totalplayers == 1 && scores[1] > 21){dealerhit=0;
329      }else
330      {
331          for(int i = 0; i < (totalplayers + 1); i++){if(scores[i] <= 21){dealerhit++;}}
332          if(dealerhit >= 1){hit(s, 0, house, cc);;}
333          if(totalplayers >= 1 ){printArray (house, "dealer ",0, bet, w);}
334          if(totalplayers >= 1 ){printArray (player, "Original ",1 , bet, w);}
335          if(totalplayers >= 2 ){printArray (split1, "Split 1 ", 2, bet, w);}
336          if(totalplayers >= 3 ){printArray (split2, "Split 2 ", 2, bet, w);}
337          if(totalplayers >= 4 ){printArray (split3, "Split 3 ", 2, bet, w);}
338          if(totalplayers >= 5 ){printArray (split4, "Split 4 ", 2, bet, w);}
339      }
340      scores[0] = cardValue(house);
341      scores[1] = cardValue(player);
342      scores[2] = cardValue(split1);
343      scores[3] = cardValue(split2);
344      scores[4] = cardValue(split3);
345      scores[5] = cardValue(split4);
346      bank = winner(scores, bank, bet, natural);
347      checkhit--;
348  }
349
350  if(cardT <= reshuffling)
351  {
352      newDeck (x, arr);
353      randomize (arr, n);
354      randomize (arr, n);
355      randomize (arr, n);
356      randomize (arr, n);
357      cardT = x;
358      cout << "Cut cards, pick a number between 1 and " << x << ": \n";
359      while(!(cin >> c))
360      {
361          cout<<"Invalid data type! Please enter 'a number' between 1 and " <<

```


Blackjack.cpp

```

x;
362         cin.clear();
363         cin.ignore(999, '\n');
364     }
365     if(c < 1 || c > x){c = 1;}
366     cutCards(arr, x, c);
367     fillStack(s, arr, x);
368     srand (time(NULL)); /* initialize random seed: */
369     reshuffling = rand() % 25 + 26; /* generate a number between 25 and 50: */
370     cout << "The Card Shoe is filled and the reshuffling card has been placed.\n\n";
371 }
372 if(bump > 0){cout << "you bumped this hand \n"; bump--;}
373 if(bump == 0){ cout << "You have " << bank << " chips. Please place your bet:";
374 while(!(cin >> bet[1]))
375     {
376         cout<<"Invalid data type! Please enter 'a number'";
377         cin.clear();
378         cin.ignore(999, '\n');
379     }
380 }
381 }
382
383 return 0;
384 }
385
386 STACK *init(int n)
387 {
388     STACK *s = (STACK *)malloc(sizeof(STACK));
389     s->top = 0;
390     s->maxelem = n;
391     s->stk = (int *)malloc(sizeof(int)*n);
392     return s;
393 }
394
395 void fillStack(STACK *s, int arr[], int x)
396 {
397     for(int i = 0; i < x; i++){
398         //cout << "top " << top << "\n";
399         //cout << "I " << i << "\n";
400         s->stk[s->top++] = arr[i];
401     }
402 }
403
404 void push(STACK *s, int data)
405 {
406     if (s == NULL) return;
407     if (s->top == s->maxelem) return;
408     s->stk[s->top++] = data;
409 }
410
411 int pop(STACK *s)

```

```

412 {
413     if (s == NULL) return -1;
414     if (s->top == 0) return -1;
415     s->top--;
416     return s->stk[s->top];
417 }
418
419 void deleteNode(STACK *s, int data)
420 {
421     if (s == NULL) return;
422     if (s->stk[s->top-1] == data) { s->top--; return; }
423     for (int i=s->top-1 ; i >= 0 ; i--)
424     {
425         if (s->stk[i] == data)
426         {
427             for ( ; i < s->top ; i++) s->stk[i] = s->stk[i+1];
428             s->top--;
429             return;
430         }
431     }
432 }
433
434 void doubleDown(STACK *s, int pno, int a[], int count[pno], double bet[pno])
435 {
436
437     int b = count[pno];
438     a[b] = pop(s);
439     count[pno]++;
440     bet[pno] = bet[pno] * 2;
441     cardT--;
442 }
443
444 void hit(STACK *s, int pno, int a[], int count[pno])
445 {
446     int b = count[pno];
447     int val = cardValue(a);
448     if(pno == 0)
449     {
450
451         while(val <= 17)
452         {
453             a[b] = pop(s);
454             cardT--;
455             count[pno]++;
456             cout << "dealer takes card number " << count[pno] << " ";
457             b = count[pno];
458             val = cardValue(a);
459         }
460         cout << "\n";
461         if(val <= 21 ){cout << "Dealer Stands.....\n";}
462         if(val > 21 ){cout << "Dealer is bust.....\n";}

```

```

463     }
464     if(pno != 0)
465     {
466     a[b] = pop(s);
467     count[pno]++;
468     cardT--;
469     }
470 }
471
472 void dealCards(STACK *s, int player[], int house[], int count[])
473 {
474     for (int i = 0; i < 2; i++)
475     {
476         player[i] = pop(s);
477         count[1]++;
478         cardT--;
479         house[i] = pop(s);
480         count[0]++;
481         cardT--;
482     }
483 }
484
485 void CheckSplit (STACK *s, int a[], int b[], int count[totalplayers], double bet[], int
splitplayer)
486 {
487     string ohand = "empty ";
488     string nhand = "empty ";
489     int split = 0;
490     int spno = splitplayer;
491     int spto = 0;
492
493     if(splitplayer == 1){ohand = "'original' ";}
494     if(splitplayer == 2){ohand = "'Split 1' ";}
495     if(splitplayer == 3){ohand = "'Split 2' ";}
496     if(splitplayer == 4){ohand = "'Split 3' ";}
497     if(splitplayer == 5){ohand = "'Split 4' ";}
498     int i = 0;
499
500     if((a[0]%13)==(a[1]%13)){split = 1;}
501     if(totalplayers == 1){spto = 2; nhand = "'Split 1' ";}
502     }else if(totalplayers == 2){spto = 3; nhand = "'Split 2' ";}
503     }else if(totalplayers == 3){spto = 4; nhand = "'Split 3' ";}
504     }else if(totalplayers == 4){spto = 5; nhand = "'Split 4' ";}
505
506     if(split == 0){splittime[0]++; i++;}
507     if(split == 1)
508     {
509         char answer;
510         cout << "would you like to split your " << ohand << "hand? ";
511         while(!(cin >> answer))
512         {

```

Blackjack.cpp

```

513         cout<<"Invalid data type! Please enter 'y' or 'n'";
514         cin.clear();
515         cin.ignore(999, '\n');
516     }
517
518     if( answer == 'y')
519     {
520
521         splittime[spto]++;
522         b[0] = a[1];
523         a[1] = 0;
524         totalplayers++;
525         count[spno]--;
526         count[spto]++;
527         bet[spto] = bet[spno];
528         hit(s, spno, a, count); //void hit(STACK *s, int pno, int
a[], int count[pno])
529         cardT--;
530     }
531 }
532 }
533
534 string cardName(int a[])
535 {
536     string ret;
537     string retb;
538     int num = 0;
539
540     for(int i = 0; a[i] != 0; i++){num++;}
541
542     for(int i = 0; i < num; i++)
543     {
544         int b = ((a[i] % 13));
545         int y = (a[i] % 52);
546         if (b == 1){ret = " Ace";}
547         if (b == 2){ret = " 2";}
548         if (b == 3){ret = " 3";}
549         if (b == 4){ret = " 4";}
550         if (b == 5){ret = " 5";}
551         if (b == 6){ret = " 6";}
552         if (b == 7){ret = " 7";}
553         if (b == 8){ret = " 8";}
554         if (b == 9){ret = " 9";}
555         if (b == 10){ret = " 10";}
556         if (b == 11){ret = " Jack";}
557         if (b == 12){ret = "Queen";}
558         if (b == 0){ret = " King";}
559         if ((y / 14) == 0){ret += "♦";}
560         if (y / 14 == 1){ret += "♠";}
561         if (y / 14 == 2){ret += "♥";}

```

Blackjack.cpp

```

562         if (y / 14 == 3){ret += "♣";}
563         if (i < num - 1){ret = ret + ", "};
564         if(num >= 1){retb += ret;}
565         else{ret = ", " + retb;}
566     }
567     return retb;
568 }
569
570 string cardNamed(int a)
571 {
572     string ret;
573
574     int b = ((a % 13));
575     int y = (a % 52);
576     if (b == 1){ret = " Ace";}
577     if (b == 2){ret = " 2";}
578     if (b == 3){ret = " 3";}
579     if (b == 4){ret = " 4";}
580     if (b == 5){ret = " 5";}
581     if (b == 6){ret = " 6";}
582     if (b == 7){ret = " 7";}
583     if (b == 8){ret = " 8";}
584     if (b == 9){ret = " 9";}
585     if (b == 10){ret = " 10";}
586     if (b == 11){ret = " Jack";}
587     if (b == 12){ret = "Queen";}
588     if (b == 0){ret = " King";}
589     if ((y / 14) == 0){ret += "♦";}
590     if (y / 14 == 1){ret += "♠";}
591     if (y / 14 == 2){ret += "♥";}
592     if (y / 14 == 3){ret += "♣";}
593
594     return ret;
595 }
596
597 int cardDealer(int a)
598 {
599     int ret = 0;
600     int aceflag = 0;
601     int b = (a % 13);
602     if (b > 10){ret += 10;}
603     if (b == 0){ret += 10;}
604     if (b == 1){ret += 11; ++aceflag;}
605     if ((b > 1) && (b <= 10)){ret = ret +b;}
606
607     return ret ;
608 }
609
610

```

Blackjack.cpp

```

611 int cardValue(int a[])
612 {
613     int ret = 0;
614     int aceflag = 0;
615     for(int i = 0; a[i] != 0; i++)
616     {
617         int b = ((a[i]) % 13);
618         if (b > 10){ret += 10;}
619         if (b == 0){ret += 10;}
620         if (b == 1){ret += 11; ++aceflag;}
621         if ((b > 1) && (b <= 10)){ret = ret +b;}
622         if ((ret > 21) && (aceflag !=0 )){ret -= 10; --aceflag;}
623     }
624     return ret ;
625 }
626
627 int winner(int a[], int bank, double bet[], int natural[])
628 {
629     int i = 1;
630     while (i <= totalplayers)
631     {
632         string hand[6]={"dealer", "Original ", "Split 1 ", "Split 2 ", "Split 3 ",
633         "Split 4 "};
634         // natural 21
635         if(natural[i] >= 1)
636         {
637             bank += (natural[i]); cout << "BlackJack Natural!! ....." << "you win
638             your bet(" << natural[i] << "). Your bank is now " << bank << " chips....\n";
639         }
640         // dealer is bust
641         if(a[0] > 21)
642         {
643             if(a[i] <= 21){bank += (bet[i]); cout << hand[i] << "
644             wins ....." << "you win your bet(" << bet[i] << "). Your bank is now " << bank << "
645             chips....\n";}
646             cout << "\n";
647         }
648         //player bumps
649         if(a[i] !=0 && a[0] <=21 && a[i] <= 21 && a[0] == a[i])
650         {
651             cout << hand[i] << " bumps ....." << hand[i] << ", your bet(" <<
652             (bet[i]) << ") will be put back. Your bank is now " << bank << " chips..... \n";
653             cout << "\n";
654         }
655         // player is bust
656         if(a[i] !=0 && a[i] > 21)
657         {
658             bank -= bet[i];
659             cout << hand[i] << " you are bust, you loose your bet(" <<

```

Blackjack.cpp

```

bet[i] << ") your bank is now " << bank << " chips..... \n";
657         cout << "\n";
658     }
659
660     // Dealer wins player bet
661     if(a[i] !=0 && a[0] <= 21 && a[0] > a[i] && a[i] < 21)
662     {
663         bank -= (bet[i]);
664         cout << hand[i] << " you lost, " << hand[0] << " wins, you loose
your bet(" << (bet[i]) << "). Your bank is now " << bank << " chips....\n";
665         cout << "\n";
666     }
667
668     //player wins bet
669     if(a[i] !=0 && a[i] <=21 && a[0] <= 21 && a[0] < a[i])
670     {
671         bank += (bet[i]);
672         cout << hand[i] << " wins ....." << "you win your bet(" <<
bet[i] << "). Your bank is now " << bank << " chips....\n";
673         cout << "\n";
674     }
675     i++;
676 }
677 return bank;
678 }
679
680 // A function to generate a new ordered deck of cards in arr[]
681 void newDeck (int x, int arr[x])
682 {
683     for (int i = 0; i < x; i++)
684         arr[i] = i+1 ;
685 }
686
687 // A utility function to print an array
688 void printArray (int cardArr[], string whoIs, int pno, double bet[pno], int n)
689 {
690     switch(n){
691     case 0:
692         cout << whoIs << " " << cardNamed(cardArr[0])<< ", " << " downcard"
<< "\n";
693         break;
694
695     case 1:
696         if(pno == 0){
697             cout << whoIs << " " << cardName(cardArr) << " Total: " <<
cardValue(cardArr) << "\n";
698         }
699         if(pno >= 1){
700             cout << whoIs <<"bet(" << bet[pno] << ") " << cardName(cardArr) << "
Total: " << cardValue(cardArr) << "\n";
701         }

```

```
702     break;
703 }
704 cout << "\n";
705 }
706
707 // A utility function to swap to integers
708 void swap (int *a, int *b)
709 {
710     int temp = *a;
711     *a = *b;
712     *b = temp;
713 }
714
715 // A function to generate a random permutation of arr[]
716 void randomize ( int arr[], int n )
717 {
718     srand ( time(NULL) );
719
720     // Start from the last element and swap one by one.
721     // We don't need to run for the first element that's why i > 0
722     for (int i = n-1; i > 0; i--)
723     {
724         srand (time(NULL)); /* initialize random seed: */
725         // Pick a random index from 0 to i
726         int j = rand() % (i+1);
727
728         // Swap arr[i] with the element at random index
729         swap(&arr[i], &arr[j]);
730     }
731     // cout << "\n";
732 }
733
734 // A utility function to swap to integers
735 void cutCards(int arr[], int x, int c)
736 {
737     c-=1;
738     int temp[x];
739     int j = c;
740     for(int i = 0; i < x; i++)
741     {
742         if (j >= x){j = (j) - x ;}
743
744         temp[i] = arr[j];
745         j++;
746     }
747
748     for(int i = 0; i < x; i++)
749     {
750         arr[i] = temp[i];
751     }
752 }
```



```

753
754 int checkNatural(int a[], int b[], double bet[], int pno)
755 {
756     int ret = 0;
757     if(cardValue(a) == 21)
758     {
759         int bjp = 0;
760         int bjd = 0;
761         int tens[2] = {0};
762         int aceflag[2] = {0};
763         for(int i = 0; a[i] != 0; i++)
764         {
765             int b = ((a[i]) % 13);
766             if (b > 10){tens[1] = 1;}
767             if (b == 0){tens[1] = 1;}
768             if (b == 1){++aceflag[1];}
769         }
770         for(int i = 0; a[i] != 0; i++)
771         {
772             int c = ((b[i]) % 13);
773             if (c > 10){tens[0] = 1;}
774             if (c == 0){tens[0] = 1;}
775             if (c == 1){++aceflag[0];}
776         }
777         if(aceflag[1] == 1 && tens[1] == 1){bjp = 1;}
778         if(aceflag[0] == 1 && tens[0] == 1){bjd = 1;}
779         if(bjp == 0){ret = 0; cout << "no natural \n";}
780         if(bjp > bjd){ret = bet[pno] * 1.5; cout << "You have a 'natural'
your bet is payed at 1.5 times its value! \n";}
781         if(bjp == bjd){ret = bet[pno]; cout << "You have a 'natural,' the
dealer has a 'natural' your bet is bumped! \n";}
782     }
783     return ret;
784 }
785

```