**★★☆☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：10315:Poker Hands**

**解題者：林柏樺**

**解題日期：2014年6月1日**

**題意：**

一副撲克牌有52張牌，每張牌有花色(分別以C，D，H，S來代表)，及點數（分別以2,3,4,5,6,7,8,9,T,J,Q,K,A來代表）。為了比出大小的緣故，我們定義2最小，A最大（花色則沒有大小關係）。

2個人各發給5張牌，並且依照以下的規則由高到低來分出勝負:

1. 同花順: 5張牌同一花色（flush），並且形成一個順子（straight）。

2. 4條: 4張牌同一個點數。

3. 葫蘆:3張同一點數，另2張同一點數。

4. 同花:5張牌同一花色。

5. 順子:5張牌的點數為連續的。請注意：點數T J Q K A是順子，而A 2 3 4 5並非順子。

6. 3條:3張牌同一個點數，另2張牌不同點數。

7. 2對:有2組2張牌同一個點數，另1張牌不同點數。若2個人都是two pairs，則依2組那對子的大小分高低，若仍相同，則以剩下的那張牌分高低。

8. 1對:有2張牌同一個點數，另3張牌皆不同點數。若2個人都是one pair，則依對子的大小分高低，若仍相同，則以剩下的那3張牌分高低。

9. High Card:沒有以上任何情況就屬於high card。若2個人都是high card，則依5張牌的點數大小依次分高低。

如果牌型相同就依序比點數大小。

比較2個人手上的牌，判斷是誰贏，或平手。

**題意範例：**

8H 9H TH JH QH 2D 3D 4D 5D 6D

🡺前5組屬於玩家Black，後5組屬於玩家White。

2H 3D 5S 9C KD 2C 3H 4S 8C AH

2H 3D 5S 9C KD 2D 3H 5C 9S KH

output:

Black wins. 🡺都同花但Q>6

White wins. 🡺K<A

Tie.

**解法：**

先把各自的牌的點數和花色分開，存到兩個陣列，然後把點數和花色的陣列各自排序，最後依照題意依序用if、else if來判斷即可。

**解法範例：**

已排序的點數陣列中(小到大): if:

1.同花順:a[n]+4=a[n+4]、花色(陣列內的值)都一樣。

2.4條:a[n]=a[n+4]。

3.葫蘆:判斷2種情形:22333、22233。

4.同花:花色(陣列內的值)都一樣。

5.順子: a[n]+i=a[n+i]。

6.3條:判斷3種情形:22234、23334、23444。

7.2對:判斷3種情形:22334、23344、22344。

8.1對:判斷4種情形:22345、23345、23445、23455。

9.High Card:從最後位依序比大小。

同牌型就依序比點數。

**討論：**

陣列排序

C++標準函式『sort』:

要先#include<algorithm>

2種排序方式:

sort(陣列開頭 ,陣列結尾); 由小到大

sort(陣列開頭 ,陣列結尾,自訂比較函式); 由自訂函式決定

自訂比較函式:

bool: if是想要的順序就return true;

不是就return false;

**程式：**

#include <stdio.h>

#include <iostream>

#include <algorithm>

using namespace std;

int point(char c) {

if(c >= '0' && c <= '9')

{return c-'0';}

if(c == 'A') {return 14;}

if(c == 'T') {return 10;}

if(c == 'J') {return 11;}

if(c == 'Q') {return 12;}

if(c == 'K') {return 13;}

}

char b[5], w[5], fb[5], fw[5];

int solve() {

sort(b, b+5); sort(w, w+5);

sort(fb, fb+5); sort(fw, fw+5);//小到大

int BB = 0, WW = 0;

int a1, a2, b1, b2, c1, c2, d1, d2;

// straight flush 同花順

if(fb[0] == fb[4] && b[0]+4 == b[4])

BB = b[4];

if(fw[0] == fw[4] && w[0]+4 == w[4])

WW = w[4];

//printf("straight flush %d %d\n", BB, WW);

if(BB > WW) return 1;

if(BB < WW) return -1;

if(BB>0 && WW>0) return 0; //要2者都為0

// four of a kind 4條

if(b[0] == b[3] || b[1] == b[4])

BB = b[1];

if(w[0] == w[3] || w[1] == w[4])

WW = w[1];

//printf("four of a kind %d %d\n", BB, WW);

if(BB > WW) return 1;

if(BB < WW) return -1;

if(BB && WW) return 0;

// full house 葫蘆

if(b[0] == b[2] && b[3] == b[4])

BB = b[0];

if(b[0] == b[1] && b[2] == b[4])

BB = b[2];

if(w[0] == w[2] && w[3] == w[4])

WW = w[0];

if(w[0] == w[1] && w[2] == w[4])

if(BB > WW) return 1;

WW = w[2];

//printf("full house %d %d\n", BB, WW);

if(BB < WW) return -1;

if(BB && WW) return 0;

// flush 同花

if(fb[0] == fb[4])

BB = 1;

if(fw[0] == fw[4])

WW = 1;

//printf("fush %d %d\n", BB, WW);

if(BB > WW) return 1;

if(BB < WW) return -1;

if(BB == 1 && WW == 1) {

for(int i = 4; i >= 0; i--) {

if(b[i] > w[i]) return 1;

if(w[i] > b[i]) return -1;

}

return 0;

}

// straight\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*順子

if(b[0]+1==b[1]&&b[1]+1==b[2]&&b[2]+1==b[3]&&b[3]+1==b[4])

BB = b[4];

if(w[0]+4 == w[4]&&w[1]+1==w[2]&&w[2]+1==w[3]&&w[3]+1==w[4])

WW = w[4];

//printf("straight %d %d\n", BB, WW);

if(BB > WW) return 1;

if(BB < WW) return -1;

if(BB && WW) return 0;

// three of a kind 3條

for(int i = 0; i < 3; i++)

if(b[i] == b[i+2])

BB = b[i];

for(int i = 0; i < 3; i++)

if(w[i] == w[i+2])

WW = w[i];

//printf("three of a kind %d %d\n", BB, WW);

if(BB > WW) return 1;

if(BB < WW) return -1;

if(BB && WW) return 0;

// two pairs 2對

if(b[0] == b[1] && b[2] == b[3])

a1 = b[0], b1 = b[2], c1 = b[4], BB = 1;

if(b[0] == b[1] && b[3] == b[4])

a1 = b[0], b1 = b[3], c1 = b[2], BB = 1;

if(b[1] == b[2] && b[3] == b[4])

a1 = b[1], b1 = b[3], c1 = b[0], BB = 1;

if(w[0] == w[1] && w[2] == w[3])

a2 = w[0], b2 = w[2], c2 = w[4], WW = 1;

if(w[0] == w[1] && w[3] == b[4])

a2 = w[0], b2 = w[3], c2 = w[2], WW = 1;

if(w[1] == w[2] && w[3] == b[4])

a2 = w[1], b2 = w[3], c2 = w[0], WW = 1;

//printf("two pair %d %d\n", BB, WW);

if(BB > WW) return 1;

if(BB < WW) return -1;

if(BB && WW) {

if(b1 != b2) return b1 > b2 ? 1 : -1;

if(a1 != a2) return a1 > a2 ? 1 : -1;

if(c1 > c2) return 1;

if(c1 < c2) return -1;

return 0;

}

// one pair 1對

if(b[0] == b[1])

BB = 1, a1 = b[0], b1 = b[2], c1 = b[3], d1 = b[4];

if(b[1] == b[2])

BB = 1, a1 = b[1], b1 = b[0], c1 = b[3], d1 = b[4];

if(b[2] == b[3])

BB = 1, a1 = b[2], b1 = b[0], c1 = b[1], d1 = b[4];

if(b[3] == b[4])

BB = 1, a1 = b[3], b1 = b[0], c1 = b[1], d1 = b[2];

if(w[0] == w[1])

WW = 1, a2 = w[0], b2 = w[2], c2 = w[3], d2 = w[4];

if(w[1] == w[2])

WW = 1, a2 = w[1], b2 = w[0], c2 = w[3], d2 = w[4];

if(w[2] == w[3])

WW = 1, a2 = w[2], b2 = w[0], c2 = w[1], d2 = w[4];

if(w[3] == w[4])

WW = 1, a2 = w[3], b2 = w[0], c2 = w[1], d2 = w[2];

//printf("one pair %d %d\n", BB, WW);

if(BB > WW) return 1;

if(BB < WW) return -1;

if(BB && WW) {

if(a1 != a2) return a1 > a2 ? 1 : -1;

if(d1 != d2) return d1 > d2 ? 1 : -1;

if(c1 != c2) return c1 > c2 ? 1 : -1;

if(b1 != b2) return b1 > b2 ? 1 : -1;

if(a1 != a2) return a1 > a2 ? 1 : -1;

return 0;

}

for(int i = 4; i >= 0; i--) {

if(b[i] > w[i]) return 1;

if(w[i] > b[i]) return -1;

}

return 0;

}

int main(int argc, char \*argv[]) {

int i, j, k;

char s[1024], B[5][3], W[5][3];

while(1) {

for(i = 0; i < 5; i++) {

if(scanf("%s", B[i]) == 1);

else return 0;

b[i] = point(B[i][0]);

fb[i] = B[i][1];

}

for(i = 0; i < 5; i++) {

scanf("%s", W[i]);

w[i] = point(W[i][0]);

fw[i] = W[i][1];

}

int flag = solve();

if(flag == 1)

printf("Black wins.\n");

else if(flag == -1)

printf("White wins.\n");

else

printf("Tie.\n");

}

system("PAUSE");

return EXIT\_SUCCESS;

}