```
import java.util.Random;

import java.util.Scanner;

import java.util.Arrays;

class nullpass extends Exception

{


}

interface cards

{

        String a[]=new String[5]; // string that holds all the suits

        int b[]=new int[13];// string that holds the number of the card

}

class card implements cards

{


        String d[]=new String[10];// string that holds the combination

}

class player extends card

{

        int amount;

        int raise;

        int wins;

        int hand[]=new int[10];

        String c1=new String();

        String c2=new String();

        int pass;

                player()

                {
```

```
                amount =1000;
                wins=0;
                raise=0;
            }




}
class gamecard extends card
{
        String g1=new String();
        String g2=new String();
        String g3=new String();
        String g4=new String();
        String g5=new String();
}

public class Main
{
        static int royal(int a,int b,int c,int d,int e,int f,int g)
            {
                int x[]=new int[100];
                x[0]=a;
                x[1]=b;
                x[2]=c;
                x[3]=d;
                x[4]=e;
                x[5]=f;
                x[6]=g;
                int i=0;
                Arrays.sort(x, 0, 7);

            if(x[0]==1 && x[3]==10&&x[4]==11&&x[5]==12&&x[6]==13)
                    return 1;
            return 0;

            }





        static int flush (String a, String b, String c, String d, String e,String f, String g)
    {

        String x = new String ();
        x = a + b + c + d + e + f + g;

        int i, j;
        int lol=0;
        char ch;
        int frequency ;
        for (i = 0; i < x.length (); i++)
        {
              frequency=0;
            ch = x.charAt(i);

            for (j = 0; j < x.length (); j++)
              {
                    if (ch == x.charAt(j))
                    {
                            ++frequency;
                    }

            }

        if(frequency>=5)
        {
            return 1;

        }

        }

        return 0;

    }



          static int straight(int a,int b,int c,int d,int e,int f,int g)
           {
                int x[]=new int[100];
                x[0]=a;
                x[1]=b;
                x[2]=c;
                x[3]=d;
                x[4]=e;
                x[5]=f;
                x[6]=g;

                int i=0;

                Arrays.sort(x, 0, 7);



            int n = x.length;


            n = removeDuplicates(x, n);

            int y[]=new int[10];

            // Print updated array

            for (i=0; i<n; i++)

                y[i]=x[i];

           for(i=0;i<=2;i++)

            {

                if(y[i+1]==y[i]+1 && y[i+2]==y[i+1]+1&&y[i+3]==y[i+2]+1&&y[i+4]==y[i+3]+1)

                    return y[i];

            }


                return 0;


            }

        static int removeDuplicates(int arr[], int n)

        {

            if (n == 0 || n == 1)

                return n;


            int j = 0;
        for (int i = 0; i < n-1; i++)
            if (arr[i] != arr[i+1])
                arr[j++] = arr[i];

        arr[j++] = arr[n-1];

        return j;
    }



    static int twopair(int a,int b,int c,int d,int e,int f,int g,int h,int k) // function for two ,two of a kind
        {
            int x[]=new int[100];
            int y[]=new int[100];
            x[0]=a;
            x[1]=b;
            x[2]=c;
            x[3]=d;
            x[4]=e;
            x[5]=f;
            x[6]=g;
            x[7]=h;
            x[8]=k;
            int l;
            l=countFreq2pair(x,8);
            return l;
        }
public static int countFreq2pair(int arr[], int n)     // frequency function for 2 2 pairs
```

```
{
    boolean visited[] = new boolean[n];
    int y[]=new int[30];
    int counter=0;

  int i;

    // Traverse through array elements and
    // count frequencies
    for ( i = 0; i < n; i++)
    {

        // Skip this element if already processed
        if (visited[i] == true)
            continue;

        // Count frequency
        int count = 1;
        for (int j = i + 1; j < n; j++) {
            if (arr[i] == arr[j]) {
                visited[j] = true;
                count++;
            }
        }
        y[i]=count;
    }
    for(i=0;i<=20;i++)
    {
        if(y[i]==2)
```

```
                counter=counter+1;
        }
        if(counter>=2)
            return 1;

        return 0      ;
}




      static int four(int a,int b,int c,int d,int e,int f,int g) // function for four of a kind
            {
                int x[]=new int[100];
                int y[]=new int[100];
                x[0]=a;
                x[1]=b;
                x[2]=c;
                x[3]=d;
                x[4]=e;
                x[5]=f;
                x[6]=g;
                int l;
                l=countFreq4(x,7);
                return l;

            }
public static int countFreq4(int arr[], int n)
{
        boolean visited[] = new boolean[n];
        // Traverse through array elements and
        // count frequencies
        for (int i = 0; i < n; i++)
        {

            // Skip this element if already processed
            if (visited[i] == true)
                continue;

            // Count frequency
            int count = 1;
            for (int j = i + 1; j < n; j++) {
                if (arr[i] == arr[j]) {
                    visited[j] = true;
                    count++;
                }
            }
            if(count==4)
                return arr[i];
        }
        return 0;
    }

         static int three(int a,int b,int c,int d,int e,int f,int g) // function for three of a kind
            {
                int x[]=new int[100];
```

```
                    x[0]=a;
                    x[1]=b;
                    x[2]=c;
                    x[3]=d;
                    x[4]=e;
                    x[5]=f;
                    x[6]=g;
                    int l;
                    l=countFreq(x,7);
                    return l;

                }
public static int countFreq(int arr[], int n) // frequency function for three of a kind
{
        boolean visited[] = new boolean[n];



        // Traverse through array elements and
        // count frequencies
        for (int i = 0; i < n; i++)
        {

                // Skip this element if already processed
                if (visited[i] == true)
                        continue;

                // Count frequency
                int count = 1;
```

```
            for (int j = i + 1; j < n; j++) {
                if (arr[i] == arr[j]) {
                    visited[j] = true;
                    count++;
                }
            }
            if(count>=3)
                return arr[i];
        }
        return 0;
    }


          static int two(int a,int b,int c,int d,int e,int f,int g) // function for two of a kind
            {
                int x[]=new int[100];
                int y[]=new int[100];
                x[1]=a;
                x[2]=b;
                x[3]=c;
                x[4]=d;
                x[5]=e;
                x[6]=f;
                x[7]=g;
                int i=7;
                int j=7;
                for(i=7;i>=0;i--)
                {
                    for(j=7;j>=0;j--)
```

```
                        {
                        if(x[i]==x[j] && i!=j)
                                return x[i];

                        }
                }

                return 0;

        }

static int max(int a,int b,int c,int d) // function for high card
        {
                if(a>b&&a>c&&a>d)
                        return a;
                else if(b>a&&b>c&&b>d)
                        return b;
                else if(c>a&&c>b&&c>d)
                        return c;
                else if(d>a&&d>c&&d>b)
                        return d;
                else
                        return 0;
        }
     public static void main(String[] args)
     {
Scanner in=new Scanner(System.in);
     card c=new card();
     c.a[0]="S";// initialising values for suits
```

```
            c.a[1]="H";

            c.a[2]="C";

            c.a[3]="D";

// initialising values for numbers of the card

            c.b[0]=1;// A is 1

            c.b[1]=2;

            c.b[2]=3;

            c.b[3]=4;

            c.b[4]=5;

            c.b[5]=6;

            c.b[6]=7;

            c.b[7]=8;

            c.b[8]=9;

            c.b[9]=10;

            c.b[10]=11;//j is 11

            c.b[11]=12;//q is 12

            c.b[12]=13;//k is 13

        Random rand = new Random();

        int ch;

        player p1=new player();

        player p2=new player();

        gamecard g=new gamecard();

        System.out.println("1)start another     game \n2)exit");

        ch=in.nextInt();

        int pot=0;

        String pause=new String();

        String pause1=new String();

        while(ch!=2)

{
```

```
int r1=rand.nextInt(4);// suit of first player first card

int r2=rand.nextInt(13);// number of first player first card

int r3=rand.nextInt(4);// suit of first     player second card

int r4=rand.nextInt(13);// number of first player second card

int r5=rand.nextInt(4);// suit of second player first card

int r6=rand.nextInt(13);// number second player first card

int r7=rand.nextInt(4);// suit of second player second card

int r8=rand.nextInt(13);// number of second player second card

int r9=rand.nextInt(4);// suit of first card to be revealed

int r10=rand.nextInt(13);// number of first card to be revealed

int r11=rand.nextInt(4);// suit of second card to be revealed

int r12=rand.nextInt(13);// number of second card to be revealed

int r13=rand.nextInt(4);// suit of third card to be revealed

int r14=rand.nextInt(13);// number of third card to be revealed

int r15=rand.nextInt(4);// suit of fourth card to be revealed

int r16=rand.nextInt(13);// number of fourth card to be revealed

int r17=rand.nextInt(4);// suit of fifth card to be revealed

int r18=rand.nextInt(13);// number of fifth card to be revealed



p1.c1=c.a[r1]+c.b[r2];// player one card one

p1.c2=c.a[r3]+c.b[r4];// player one card two

p2.c1=c.a[r5]+c.b[r6];// player two card one

p2.c2=c.a[r7]+c.b[r8];// player two card two

g.g1=c.a[r9]+c.b[r10];// game card 1

g.g2=c.a[r11]+c.b[r12];// game card 2

g.g3=c.a[r13]+c.b[r14];// game card 3

g.g4=c.a[r15]+c.b[r16];// game card 4

g.g5=c.a[r17]+c.b[r18];// game card 5
```

```
System.out.println("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
System.out.println("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
System.out.println("\t\t\t\t reveal player ones cardds\n");
System.out.println("\nplayer one card one\t"+p1.c1);
System.out.println("\nplayer one card two\t"+p1.c2);
System.out.println("press any key to continue");
pause1=in.next();
System.out.println("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
System.out.println("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
System.out.println("\t\t\t reveal player two's cards\n");
System.out.println("\nplayer two card one\t"+p2.c1);
System.out.println("\nplayer two card two\t"+p2.c2);
System.out.println("press any key to continue");
pause=in.next();
System.out.println("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
System.out.println("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
System.out.println("\t\t\treveal game cards one two and three");
System.out.println("\n\t\t\tone\t"+g.g1);
System.out.println("\n\t\t\ttwo\t"+g.g2);
System.out.println(" \n\t\t\tthree\t"+g.g3);
System.out.println("press any key to continue");
pause=in.next();
System.out.println("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
System.out.println("\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n\n");
System.out.println("\n\t\t\tplace minimum bets\t\t\t");
p1.amount=p1.amount-5;
p2.amount=p2.amount-5;
pot=10;
System.out.println("\n \t\t\tplayer one's turn");
System.out.println("1)raise \n2)pass\n3)fold");

int ch1;

p1.pass=0;

p2.pass=0;

ch1=in.nextInt();

if(ch1==1)

{

        System.out.println("enter the amount of money to raise by");

        p1.raise=in.nextInt();

        p1.amount=p1.amount-p1.raise;

        pot=pot+p1.raise;


}

else if(ch1==2)

{       p1.raise=0;

        System.out.println("player one has passed this round\n");

        p1.pass=1;

}

else if(ch1==3)

{

        p2.wins++;

        continue;

}

System.out.println("\n \t\t\tplayer two's turn");

System.out.println("1)raise \n2)call\n3)pass\n4)fold");

int ch2;

ch2=in.nextInt();

if(ch2==1)

{
		System.out.println("enter the amount of money to raise by");
		p2.raise=in.nextInt();
		p2.amount=p2.amount-p2.raise;
		pot=pot+p2.raise;


	}
	else if(ch2==2)
	{
		p2.raise=p1.raise-p2.raise;
		p2.amount=p2.amount-p2.raise;
		pot=pot+p2.raise;


	}
	else if(ch2==3)
	{
		try
		{
			if(p1.pass==1)
			{
				System.out.println("player two has passed this round");
				p2.raise=0;
			}
			else
				throw new nullpass();
		}
		catch(nullpass e)
		{
			System.out.println("cant pass");
			System.out.println("\n \t\t\tplayer two's turn");
```

```
                System.out.println("1)raise \n2)call");
                ch2=in.nextInt();
                if(ch2==1)
                {
                        System.out.println("enter the amount of money to raise by");
                        p2.raise=in.nextInt();
                        p2.amount=p2.amount-p2.raise;
                        pot=pot+p2.raise;

                }
                else if(ch2==2)
                {
                        p2.raise=p1.raise-p2.raise;
                        p2.amount=p2.amount-p2.raise;
                        pot=pot+p2.raise;

                }
        }

}
else if(ch2==4)
{
        p1.wins++;
        continue;
}
if(p1.raise!=p2.raise)
{
        if(p1.raise>p2.raise)
        {
                System.out.println("\n\t\t\t player two's move");
                System.out.println("\n \t\t\t 1)call\n \t\t\t2)fold");
                ch2=in.nextInt();
                if(ch2==1)
                {
                        p2.raise=p1.raise-p2.raise;
                        p2.amount=p2.amount-p2.raise;
                        pot=pot+p2.raise;
                }

        }
          else if(p1.raise<p2.raise)
        {
                System.out.println("\n\t\t\t player one's move");
                System.out.println("\n \t\t\t 1)call\n \t\t\t2)fold");
                ch1=in.nextInt();
                if(ch1==1)
                {
                        p1.raise=p2.raise-p1.raise;
                        p1.amount=p1.amount-p1.raise;
                        pot=pot+p1.raise;
                }

        }
}
p1.raise=0;
p2.raise=0;
p1.pass=0;
System.out.println("\n\t\t\tamount left after round for player one\t"+p1.amount);
System.out.println("\n\t\t\tamount left after round for player two\t"+p2.amount);
System.out.println("the pot has"+pot);


 System.out.println("\n\n\n\n\n\n\t\t\t four\t"+g.g4);


System.out.println("\n \t\t\tplayer one's turn");
System.out.println("1)raise \n2)pass \n3)fold");
ch1=in.nextInt();
if(ch1==1)
{
        System.out.println("enter the amount of money to raise by");
        p1.raise=in.nextInt();
        p1.amount=p1.amount-p1.raise;
        pot=pot+p1.raise;


}
else if(ch1==2)
{
        p1.raise=0;
        System.out.println("player one has passed this round\n");
        p1.pass=1;
}
else if(ch1==3)
{
        p2.wins++;
        continue;
}
System.out.println("\n \t\t\tplayer two's turn");
System.out.println("1)raise \n2)call\n3)pass\n4)fold");
```

```
ch2=in.nextInt();
if(ch2==1)
{
        System.out.println("enter the amount of money to raise by");
        p2.raise=in.nextInt();
        p2.amount=p2.amount-p2.raise;
        pot=pot+p2.raise;

}
else if(ch2==2)
{
        p2.raise=p1.raise-p2.raise;
        p2.amount=p2.amount-p2.raise;
        pot=pot+p2.raise;

}
else if(ch2==3)
{
        try
        {
                if(p1.pass==1)
                {
                        System.out.println("player two has passed this round");
                        p2.raise=0;
                }
                else
                        throw new nullpass();
        }
        catch(nullpass e)
        {
                System.out.println("cant pass");
                System.out.println("\n \t\t\tplayer two's turn");
                System.out.println("1)raise \n2)call");
                ch2=in.nextInt();
                if(ch2==1)
                {
                        System.out.println("enter the amount of money to raise by");
                        p2.raise=in.nextInt();
                        p2.amount=p2.amount-p2.raise;
                        pot=pot+p2.raise;

                }
                else if(ch2==2)
                {
                        p2.raise=p1.raise-p2.raise;
                        p2.amount=p2.amount-p2.raise;
                        pot=pot+p2.raise;

                }
        }

}
else if(ch2==4)
{
        p1.wins++;
        continue;
}
if(p1.raise!=p2.raise)
```

```
{
        if(p1.raise>p2.raise)
        {
                System.out.println("\n\t\t\t player two's move");
                System.out.println("\n \t\t\t 1)call\n \t\t\t2)fold");
                ch2=in.nextInt();
                if(ch2==1)
                {
                        p2.raise=p1.raise-p2.raise;
                        p2.amount=p2.amount-p2.raise;
                        pot=pot+p2.raise;
                }

        }
          else if(p1.raise<p2.raise)
        {
                System.out.println("\n\t\t\t player one's move");
                System.out.println("\n \t\t\t 1)call\n \t\t\t2)fold");
                ch1=in.nextInt();
                if(ch1==1)
                {
                        p1.raise=p2.raise-p1.raise;
                        p1.amount=p1.amount-p1.raise;
                        pot=pot+p1.raise;
                }

        }
}
p1.raise=0;
```

```
p2.raise=0;
p1.pass=0;
System.out.println("\n\t\t\tamount left after round for player one"+p1.amount);
System.out.println("\n\t\t\tamount left after round for player two"+p2.amount);
System.out.println("the pot has"+pot);

    System.out.println("\n\n\n\n \t\t\t five\t"+g.g5);

System.out.println("\n \t\t\tplayer one's turn");
System.out.println("1)raise \n2)pass \n3)fold");
ch1=in.nextInt();
if(ch1==1)
{
        System.out.println("enter the amount of money to raise by");
        p1.raise=in.nextInt();
        p1.amount=p1.amount-p1.raise;
        pot=pot+p1.raise;

}
else if(ch1==2)
{
        p1.raise=0;
        System.out.println("player one has passed this round\n");
        p1.pass=1;
}
else if(ch1==3)
{
        p2.wins++;
        continue;
```

```
}
System.out.println("\n \t\t\tplayer two's turn");
System.out.println("1)raise \n2)call\n3)pass\n4)fold");

ch2=in.nextInt();
if(ch2==1)
{
        System.out.println("enter the amount of money to raise by");
        p2.raise=in.nextInt();
        p2.amount=p2.amount-p2.raise;
        pot=pot+p2.raise;

}
else if(ch2==2)
{
        p2.raise=p1.raise-p2.raise;
        p2.amount=p2.amount-p2.raise;
        pot=pot+p2.raise;

}
else if(ch2==3)
{
try
        {
                if(p1.pass==1)
                {
                        System.out.println("player two has passed this round");
                        p2.raise=0;
                }
        else
            throw new nullpass();
    }
    catch(nullpass e)
    {
        System.out.println("cant pass");
        System.out.println("\n \t\t\tplayer two's turn");
        System.out.println("1)raise \n2)call\n");
        ch2=in.nextInt();
        if(ch2==1)
        {
            System.out.println("enter the amount of money to raise by");
            p2.raise=in.nextInt();
            p2.amount=p2.amount-p2.raise;
            pot=pot+p2.raise;

        }
        else if(ch2==2)
        {
            p2.raise=p1.raise-p2.raise;
            p2.amount=p2.amount-p2.raise;
            pot=pot+p2.raise;

        }
    }

}
else if(ch2==4)
{
        p1.wins++;
        continue;
}
if(p1.raise!=p2.raise)
{
        if(p1.raise>p2.raise)
        {
                System.out.println("\n\t\t\t player two's move");
                System.out.println("\n \t\t\t1)call\n\t\t\t2)fold");
                ch2=in.nextInt();
                if(ch2==1)
                {
                        p2.raise=p1.raise-p2.raise;
                        p2.amount=p2.amount-p2.raise;
                        pot=pot+p2.raise;
                }

        }
          else if(p1.raise<p2.raise)
        {
                System.out.println("\n\t\t\t player one's move");
                System.out.println("\n\t\t\t1)call\n \t\t\t2)fold");
                ch1=in.nextInt();
                if(ch1==1)
                {
                        p1.raise=p2.raise-p1.raise;
                        p1.amount=p1.amount-p1.raise;
                        pot=pot+p1.raise;
                }
        }
    }
    p1.raise=0;
    p2.raise=0;
    System.out.println("\n\t\t\tamount left after round for player one"+p1.amount);
    System.out.println("\n\t\t\tamount left after round for player two"+p2.amount);
    System.out.println("the pot has"+pot);
    int i;
    for(i=0;i<=9;i++)
    {
        p1.hand[i]=0;
        p2.hand[i]=0;
    }
    int h;
    int m;
    int o,p,q;
    //1) high card conditions

    h=Main.max(c.b[r2],c.b[r4],c.b[r6],c.b[r8]);// h has the max value
    if(h==c.b[r2]||h==c.b[r3])
        p1.hand[0]=1;
    else
        p2.hand[0]=1;

//    pair
        h=Main.two(c.b[r2],c.b[r4],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
      m=Main.two(c.b[r6],c.b[r8],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
      if(h>m)
```

```
                p1.hand[1]=1;
        if(m>h)
                p2.hand[1]=1;



// 3 of a kind
h=Main.three(c.b[r2],c.b[r4],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
m=Main.three(c.b[r6],c.b[r8],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
        if(h>m)
                p1.hand[2]=1;
        else if(m>h)
                p2.hand[2]=1;

    // straight
          h=Main.straight(c.b[r2],c.b[r4],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
  m=Main.straight(c.b[r6],c.b[r8],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
        if(h>m)
                p1.hand[3]=1;
        else if(m>h)
                p2.hand[3]=1;





//flush
         h =Main.flush (c.a[r1], c.a[r3], c.a[r9], c.a[r11], c.a[r13], c.a[r15],c.a[r17]);
         m =Main.flush (c.a[r3], c.a[r5], c.a[r9], c.a[r11], c.a[r13], c.a[r15],c.a[r17]);
p1.hand[7]=0;
p2.hand[7]=0;
```

```
    if (h==1)
                p1.hand[4] = 1;
    if (m==1)
                p2.hand[4] = 1;

     // full house


     //pair
    h=Main.two(c.b[r2],c.b[r4],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
    m=Main.two(c.b[r6],c.b[r8],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);

    // 3 of a kind
    q=Main.three(c.b[r2],c.b[r4],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
    o=Main.three(c.b[r6],c.b[r8],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
          if(h!=q &&h!=0 && q!=0)
                p1.hand[5]=1;
          if(m!=o && m!=0 && 0!=0)
                p2.hand[5]=1;




// 4 of a kind
     h=Main.four(c.b[r2],c.b[r4],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
     m=Main.four(c.b[r6],c.b[r8],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
          if(h>m)
                p1.hand[6]=1;
          else if(m>h)
                p2.hand[6]=1;
```

```
// straight flush

    //straight

          h=Main.straight(c.b[r2],c.b[r6],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
          m=Main.straight(c.b[r6],c.b[r8],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);


 //flush
         q =Main.flush (c.a[r1], c.a[r3], c.a[r9], c.a[r11], c.a[r13], c.a[r15],c.a[r17]);
         o =Main.flush (c.a[r5], c.a[r7], c.a[r9], c.a[r11], c.a[r13], c.a[r15],c.a[r17]);

         if(h!=0 && q==1)
              p1.hand[7]=1;
         if(m!=1 && o==1)
              p2.hand[7]=1;


           //royal flush
          // royal
         h=Main.royal(c.b[r2],c.b[r4],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
         m=Main.royal(c.b[r6],c.b[r8],c.b[r10],c.b[r12],c.b[r14],c.b[r16],c.b[r18]);
         // flush
         q =Main.flush (c.a[r1], c.a[r3], c.a[r9], c.a[r11], c.a[r13], c.a[r15],c.a[r17]);
```

```
     o =Main.flush (c.a[r5], c.a[r7], c.a[r9], c.a[r11], c.a[r13], c.a[r15],c.a[r17]);
if(h==1&&q==1)
        p1.hand[8]=1;
if(m==1&&o==1)
        p2.hand[8]=1;




for(i=8;i>=0;i--)
{
        if(p1.hand[i]>p2.hand[i])
        {
                System.out.println("winner is p1");
                p1.wins++;
                p1.amount=p1.amount+pot;
                p1.raise=0;
                break;
        }
          else if(p1.hand[i]<p2.hand[i])
        {
                System.out.println("winner is p2");
                p2.wins++;
                p2.amount=p2.amount+pot;
                p2.raise=0;
                break;
        }
```

```
            }

        System.out.println("1)start another    game \n2)exit");
        ch=in.nextInt();

    }

           }
    }
```