```
;Package assignments

;import java.util.Scanner


} Public class PermutationsCombinationsProbability

        } Public static void main(String[] args)
                                                } try
        ;Scanner input = new Scanner(System.in)

 System.out.println("Enter {n} number:\nMust be
             ;greater than zero and greater than 6.")
                        ;()int n = input.nextInt

 System.out.println("Enter {r} number:\nMust be
     ;greater than zero and not exceeding {n} value")
                        ;()int r = input.nextInt

                ;long per = r_Permutation(n, r)
               ;long comb = r_Combination(n, r)
                    ;long prob = probability(n)

System.out.println("R-Permutation for [ " + n +
                   ;", " + r + " ] Is: {" + per + "}")
System.out.println("R-Combination for [ " + n +
                  ;", " + r + " ] Is: {" + comb + "}")
System.out.println("Probability for [ " + n + "
                               ;] is: {" + prob + "}")

            } catch (IllegalArgumentException i) {
          System.out.println("Input Error: " +
                                    ;i.getMessage())
                           } catch (Exception e) {
     System.out.println("Unexpected Error: " +
                                    ;e.getMessage())
                                              {
                                                {


            } Public static long factorial(int num)
                                  } if (num < 0)
       throw new IllegalArgumentException("The
       ;factorial Is undefined for negative numbers!")
                                              {
                                  ;long result = 1
                } for (int i = 2; i <= num; i++)
                                    ;result *= i
                                              {
                                    ;return result
                                                {

} Public static boolean Conditions(int n, int r)
                } if (n <= 0 || r <= 0 || n < r)
```

```
throw new IllegalArgumentException("[ " + n + ",
  " + r + " ] values must be greater than 0, and r must
                                        ;not exceed n.")
                                                    {
                                        ;return true
                                                      {

     }Public static long r_Permutation(int n, int r)
                                    ;Conditions(n, r)
            ;return factorial(n) / factorial(n - r)
                                                      {

     }Public static long r_Combination(int n, int r)
                                    ;Conditions(n, r)
 return factorial(n) / (factorial(r) * factorial(n
                                                  ;- r))
                                                     {

             }Public static long probability(long n)
                                        }if (n > 6)
                                  return 6 / n;
                                          }else{
throw new IllegalArgumentException("The value of
               ;n (" + n + ") must be greater than 6!")
                                                     {
                                                       {
                                                         {
```