```java
package math.number_theory;

public class ExtendedEuclid {

  /*
   * Extended Euclid Algorithm for Solving Linear Diophantine Equation
   * Solve: ax + by = c. Let d = gcd(a, b).
   * If d | c, then it has infinite number of solutions. Otherwise, it has no solution.
   * The following algorithm derives one solution (x0, y0). Other solutions can be drived by:
   * x = x0 + n * (b / d), y = y0 - n * (a / d)
   */

  static int x, y, d;

  void extendedEuclid(int a, int b)
  {
    if(b == 0) { x = 1; y = 0; d = a; return; }
    extendedEuclid(b, a % b);
    int x1 = y;
    int y1 = x - a / b * y;
    x = x1; y = y1;
  }
}
```

```cpp
// find n! mod p
int factmod (ll n, ll p) {
    ll res = 1;
    while (n > 1) {
        res = (res * ((n/p) % 2ll ?(p-1) : 1ll)) % p;
        for (ll i=2; i<=n%p; ++i)
            res = (res * i) % p;
        n /= p;
    }
    return res % p;
}
```