```
/* Rollar Rho */

#define mod 8051
int f(int x){
  return (x*x + 1)%mod;
}
int buscar_factor(int n) {
  int x = 2, y = 2, d = 1;
  while(d == 1) {
    x = f(x);
    y = f(f(y));
    d = __gcd(abs(x - y), n);
  }
  return d;
}

/* Rotar Matrix */

template<class T>
struct Matrix {
  T **mat;
  int fila, columna;
  Matrix(int f = 3, int c = 3) {
    mat = new T*[f];
    for(int i = 0; i < f; i++)
      mat[i] = new T[c];
    fila = f; columna = c;
  }
  void normalizar(int &x){
    x = x % 4; x = (x + 4) % 4;
  }
  void Rotar(int veces) {
    normalizar(veces);
    for(int t = 0; t < veces; t++) {
      Matrix N(fila, columna);
      for (int i = 0; i < fila; i++)
        for (int j = 0; j < columna; j++)
          N.mat[i][j] = mat[i][j];
      for(int i = 0; i < fila; i++)
        delete []mat[i];
      delete []mat;
      mat = new T*[N.columna];
      for(int i = 0; i < N.columna; i++)
        mat[i] = new T[N.fila];
      fila = N.columna; columna = N.fila;
      int i = 0, k = columna - 1;
      for (; i < columna; i++, k--) {
        for (int j = 0; j < fila; j++)
          mat[j][k] = N.mat[i][j];
} } } };
// Matrix<int> A;
```