```
/* This program compute the inverse, x, of a and n (0<a<n) such that
ax (mod n) = 1 */

#include <stdio.h>
main()
{
   int g[100], u[100], v[100];   /* Initialize arrays for indexing */

   int i=1;                      /* Beginning index # of loop */
   int y,n,a;                    /* Defining input and intermediate var. */
   printf ("inverse of a,n. Enter a,n separated by space: ");
   scanf  ("%d %d0", &a, &n);    /* Read in a and n */
   g[0]= n;
   g[1]= a;
   u[0]= v[1] = 1;
   u[1] = v[0] = 0;
   while (g[i])
   {
      g[i]= u[i] * n + v[i] * a;
      y= g[i-1]/g[i];
      g[i+1] = g[i-1] - y*g[i];
      u[i+1] = u[i-1] - y*u[i];
      v[i+1] = v[i-1] - y*v[i];
      i++;
   }                             /* Using extension of Euclid's gcd algo */
   if (v[i-1] <= 0)
   {
      printf ("inv of %d and %d is %d \n", a,n,v[i-1]+n);
   }
   else
   {
      printf ("inv of %d and %d is %d \n",a,n,v[i-1]+2*n);
   }
}
```

```
/*
This program uses Euclid's algorithm to solve for the greatest common
denominator (gcd) of two number. Given two input integers, a and n, this
program provides their mutual gcd. This is intended to be an example for
generating keys in the RSA public key system */

#include <stdio.h>
main()
{
   int   g[100];     /* Initialize an array for gcd */
   int   i=1;
   printf ("gcd of a,n. Enter a,n separated by space:");
   scanf ("%d %d0", &g[0], &g[1]);
   while (g[i])
   {
     g[i+1] = g[i-1] % g[i];
     i++;
   }
   printf ("gcd of %d and %d is %d \n",g[0],g[1],g[i-1]);
}

**********************************************************************
```