```
 1: #include <stdlib.h>
 2: #include <stdio.h>
 3: #include <math.h>
 4:
 5: #include "GeraGrafo.h"
 6:
 7: void imprimeErro ( char* exename )
 8: {
 9:     printf( "Gerador de grafos. Uso: %s [nvertex tipo(0|1) [debuglevel(0-2)]]\n", exename );
10:     printf( "Se chamado sem parametros, gera 20 sequencias, 10 de cada tipo.\n" );
11:     printf( "Tipo 0: p = 0.2\n" );
12:     printf( "Tipo 1: p = 1 / ( 2 * ( n ^ 0.5 ) )\n" );
13: }
14:
15: int main ( int argc, char ** argv )
16: {
17:     int debug = 0;
18:     int nVertex = 0;
19:
20:     if ( argc < 3 )
21:     {
22:         for ( int c = 0; c < NSEQS; c++ )
23:         {
24:             for ( int i = 1; i <= NGRAFOS_POR_SEQ; i++ )
25:             {
26:                 char path[ 256 ];
27:
28:                 sprintf( path, "grafo_p%d_%02d.grafo", c, i );
29:
30:                 FILE * f = fopen( path , "w" );
31:
32:                 int n = 10 * ( ( int ) pow( 2, i ) );
33:
34:                 // gera grafo com debug = 1 (so para dizer o que esta fazendo, nao depurar)
35:                 GeraGrafo( f, n, c, 1 );
36:
37:                 fclose( f );
38:             }
39:         }
40:     }
41:     else
42:     {
43:         nVertex = ( int ) ( atoi( argv[1] ) );
44:
45:         int type = atoi( argv[2] );
46:         if ( type > 1 )
47:         {
48:             printf( "Tipo invalido.\n\n" );
49:             imprimeErro( argv[0] );
50:
51:             exit( 2 );
52:         }
53:
54:         if ( nVertex <= 0 || nVertex > 100000 )
55:         {
56:             printf( "Numero de vertices invalido (%d).\n\n", nVertex );
57:             imprimeErro( argv[0] );
58:
59:             exit( 3 );
60:         }
61:
62:         if ( argc > 3 )
63:         {
64:             debug = atoi( argv[3] );
65:         }
66:
67:         GeraGrafo( stdout, nVertex, type, debug );
68:     }
69:
70:     return 1;
71: }
```