

**NAME**

gvgen – generate graphs

**SYNOPSIS**

**gvgen** [ **-dv?** ] [ **-in** ] [ **-cn** ] [ **-C***x,y* ] [ **-g**/*f*/*x,y* ] [ **-G**/*f*/*x,y* ] [ **-hn** ] [ **-kn** ] [ **-b***x,y* ] [ **-B***x,y* ] [ **-mn** ] [ **-M***x,y* ] [ **-pn** ] [ **-r***x,y* ] [ **-R***x* ] [ **-sn** ] [ **-Sn** ] [ **-S***n,d* ] [ **-tn** ] [ **-td***n* ] [ **-T***x,y* ] [ **-T***x,y,u,v* ] [ **-wn** ] [ **-n***prefix* ] [ **-N***name* ] [ **-o***outfile* ]

**DESCRIPTION**

**gvgen** generates a variety of simple, regularly-structured abstract graphs.

**OPTIONS**

The following options are supported:

- c** *n*     Generate a cycle with *n* vertices and edges.
- C** *x,y*   Generate an *x* by *y* cylinder. This will have *x*\**y* vertices and 2\**x*\**y* - *y* edges.
- g** *f*/*x,y*   Generate an *x* by *y* grid. If *f* is given, the grid is folded, with an edge attaching each pair of opposing corner vertices. This will have *x*\**y* vertices and 2\**x*\**y* - *y* - *x* edges if unfolded and 2\**x*\**y* - *y* - *x* + 2 edges if folded.
- G** *f*/*x,y*   Generate an *x* by *y* partial grid. If *f* is given, the grid is folded, with an edge attaching each pair of opposing corner vertices. This will have *x*\**y* vertices.
- h** *n*     Generate a hypercube of degree *n*. This will have 2<sup>*n*</sup> vertices and *n*\*2<sup>(*n*-1)</sup> edges.
- k** *n*     Generate a complete graph on *n* vertices with *n*\*(*n*-1)/2 edges.
- b** *x,y*   Generate a complete *x* by *y* bipartite graph. This will have *x*+*y* vertices and *x*\**y* edges.
- B** *x,y*   Generate an *x* by *y* ball, i.e., an *x* by *y* cylinder with two "cap" nodes closing the ends. This will have *x*\**y* + 2 vertices and 2\**x*\**y* + *y* edges.
- m** *n*     Generate a triangular mesh with *n* vertices on a side. This will have (*n*+1)\**n*/2 vertices and 3\*(*n*-1)\**n*/2 edges.
- M** *x,y*   Generate an *x* by *y* Moebius strip. This will have *x*\**y* vertices and 2\**x*\**y* - *y* edges.
- p** *n*     Generate a path on *n* vertices. This will have *n*-1 edges.
- r** *x,y*   Generate a random graph. The number of vertices will be the largest value of the form 2<sup>*n*</sup>-1 less than or equal to *x*. Larger values of *y* increase the density of the graph.
- R** *x*     Generate a random rooted tree on *x* vertices.
- s** *n*     Generate a star on *n* vertices. This will have *n*-1 edges.
- S** *n*     Generate a Sierpinski graph of order *n*. This will have 3\*(3<sup>(*n*-1)</sup> + 1)/2 vertices and 3<sup>*n*</sup> edges.
- S** *n,d*   Generate a *d*-dimensional Sierpinski graph of order *n*. At present, *d* must be 2 or 3. For *d* equal to 3, there will be 4\*(4<sup>(*n*-1)</sup> + 1)/2 vertices and 6 \* 4<sup>(*n*-1)</sup> edges.
- t** *n*     Generate a binary tree of height *n*. This will have 2<sup>*n*</sup>-1 vertices and 2<sup>*n*</sup>-2 edges.
- t** *h,n*   Generate a *n*-ary tree of height *h*.
- T** *x,y*   Generate an *x* by *y* torus. This will have *x*\**y* vertices and 2\**x*\**y* edges. If *u* and *v* are given, they specify twists of that amount in the horizontal and vertical directions, respectively.
- w** *n*     Generate a path on *n* vertices. This will have *n*-1 edges.
- i** *n*     Generate *n* graphs of the requested type. At present, only available if the **-R** flag is used.

- n** *prefix*  
Normally, integers are used as node names. If *prefix* is specified, this will be prepended to the integer to create the name.
- N** *name*  
Use *name* as the name of the graph. By default, the graph is anonymous.
- o** *outfile*  
If specified, the generated graph is written into the file *outfile*. Otherwise, the graph is written to standard out.
- d**  
Make the generated graph directed.
- v**  
Verbose output.
- ?**  
Print usage information.

**EXIT STATUS**

**gvgen** exits with 0 on successful completion, and exits with 1 if given an ill-formed or incorrect flag, or if the specified output file could not be opened.

**AUTHOR**

Emden R. Gansner <erg@research.att.com>

**SEE ALSO**

gc(1), acyclic(1), gvpr(1), gvcolor(1), ccomps(1), sccmap(1), tred(1), libgraph(3)