

1. Making a graph. I'm just creating a file `/tmp/board,a,b,c,d,e,f,g.gb`, where *a* thru *g* appear on the command line.

```
#include "gb_graph.h"      /* we use the GB_GRAPH data structures */
#include "gb_basic.h"      /* and the basic graph operations */
#include "gb_save.h"       /* and we save our results in ASCII format */
long a, b, c, d, e, f, z;
char buf[100];
int main(int argc, char *argv[])
{
    Graph *g, *gg, *ggg;
    if (argc != 8 ∨ sscanf(argv[1], "%ld", &a) != 1 ∨ sscanf(argv[2], "%ld", &b) != 1 ∨ sscanf(argv[3], "%ld",
        &c) != 1 ∨ sscanf(argv[4], "%ld", &d) != 1 ∨ sscanf(argv[5], "%ld", &e) != 1 ∨ sscanf(argv[6],
        "%ld", &f) != 1 ∨ sscanf(argv[7], "%ld", &z) != 1) {
        fprintf(stderr, "Usage: %s a b c d e f g\n", argv[0]);
        exit(-1);
    }
    g = board(a, b, c, d, e, f, z);
    sprintf(buf, "/tmp/board,%ld,%ld,%ld,%ld,%ld,%ld,%ld.gb", a, b, c, d, e, f, z);
    save_graph(g, buf);      /* generate an ASCII file for it */
    return 0;                /* normal exit */
}
```

2. Index.*a*: 1.*argc*: 1.*argv*: 1.*b*: 1.*board*: 1.*buf*: 1.*c*: 1.*d*: 1.*e*: 1.*exit*: 1.*f*: 1.*fprintf*: 1.*g*: 1.*gg*: 1.*ggg*: 1.**Graph**: 1.*main*: 1.*save_graph*: 1.*sprintf*: 1.*sscanf*: 1.*stderr*: 1.*z*: 1.

MAKEBOARD

	Section	Page
Making a graph	1	1
Index	2	2