```
#define MAX_TPC 4 // Threads per core (max SMT mode is SMT4)
#define MAX_CPS 22 // Cores per socket
#define MAX_SPS 2
#define MAX_THR (MAX_TPC * MAX_CPS * MAX_SPS)

void Print_Map(int sps, int cps, int tpc, int base) {
  const int maps[MAX_TPC][MAX_TPC] = {
    { 0                      },
    { 0,          4          },
    { 0,    2,    4          },
    { 0,    2,    4,    6    },
    { 0, 1, 2,    4,    6    },
    { 0, 1, 2,    4, 5, 6    },
    { 0, 1, 2, 3, 4, 5, 6    },
    { 0, 1, 2, 3, 4, 5, 6, 7 }
  };

  const int sep = ',';

  int thread, core, socket;

  int tot = sps * cps * tpc;
  int cur = 0;

  for (socket = 0; socket < sps; ++socket) {
    for (core = 0; core < cps; ++core) {
      for (thread = 0; thread < tpc; ++thread) {
        int shift = socket * MAX_CPS * MAX_TPC +
                        core * MAX_TPC;
        shift += base;
        ++cur;
        int c = (cur != tot) ? sep : '\n';
        printf("%d%c", shift + maps[tpc-1][thread], c);
      }
    }
  }
}

void Print_Usage(char **argv) {
    fprintf(stderr, "Usage: %s "
        "threads_per_core=[1-%d] "
        "cores_per_socket=[1-%d] "
        "sockets_per_system=[1-%d] "
        "base_thread=[0-%d]\n",
        argv[0], MAX_TPC, MAX_CPS, MAX_SPS, MAX_THR-1);
}

int main(int argc, char **argv) {
  const int num_args = 4;

  if (argc != num_args+1) {
    fprintf(stderr, "Invalid number of arguments (%d). Expecting %d "
        "arguments.\n", argc-1, num_args);
    Print_Usage(argv);
    exit(EXIT_FAILURE);
```