```c
/*  Copyleft Bromley brom@physics.uq.edu.au August 2018
 *  Implementing a couple of the simplest random methods in c

 *  basic compile eg. with:
 *  gcc -Wall -Wextra randomised-simple.c -o randomised-simple.exe -lm
 *  then test usage eg.:
 *  ./randomised-simple.exe 12345 10
 *
 *  older gnu compilers might need -std=gnu99 for drand48 support
 *
 *  ************80 char length page width best for a2ps ***********************
 */

#include <stdlib.h>
#include <stdio.h>
#include <math.h>

int main(int argc, char *argv[]) {
  unsigned int iseed;  /* input 1 random number seed */
  int nloops;          /* input 2 number of loops */

  int jloops;

  int irandcur;
  double drandcur;

  if (argc != 3) {
    fprintf(stderr, "Usage: %s <iseed> <nloops>\n", argv[0]);
    exit(EXIT_FAILURE);
  }

  iseed = atoi(argv[1]);
  nloops = atoi(argv[2]);

  printf("Running %s with iseed=%u and nloops=%d \n", argv[0],iseed,nloops);

/* method 1 rand() generation.... see ** man rand **
 * RAND_MAX is defined in stdlib.h
 * First generate the seed and then run off nloops*/
  printf("method1 using rand() with RAND_MAX=%d \n", RAND_MAX);
  srand(iseed);
  for (jloops = 0; jloops < nloops; jloops++) {
    irandcur =  rand();
    drandcur = (double)rand() / ((double)RAND_MAX+1.0);
    printf("method1 int=%d, dble=%f, \n", irandcur,drandcur);
  }

  printf("\n");

/* method 2 drand48() generation (uses long int seed)
 * ** man drand48 **
 * for more information about (lack of) thread safety */
  double drandcur1,drandcur2;
  long int iseedlong;

  printf("method2 using drand48()");
  iseedlong = (long int) iseed;
  srand48(iseedlong);
  for (jloops = 0; jloops < nloops; jloops++) {
    drandcur1 = drand48(); // random number between 0,1
    drandcur2 = drand48(); // random number between 0,1
    printf("method2 drandcur1=%g, drandcur2=%g,\n",drandcur1,drandcur2);
  }

  exit(EXIT_SUCCESS);
}
```