```
//-------------------------------------------------------------------------
// c c o v e r a g e   --   calculate the coverage of feature detectors
//-------------------------------------------------------------------------
// COMPILE: gcc -Wall -g -o ccoverage ccoverage.c -lm
//-------------------------------------------------------------------------

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

#define MAXFN   200
#define MAXLINE 2000
#define MAXLOCS 200000

int
load_locations (char *fn, float locs[][2], int first)
{
  char line[MAXLINE], *kwd;
  int nlocs;
  FILE *f;

  if ((f = fopen (fn, "r")) == NULL) {
    fprintf (stderr, "Cannot open %s!\n", fn);
    exit (EXIT_FAILURE);
  }
  fgets (line, MAXLINE, f);
  fgets (line, MAXLINE, f);
  nlocs = first;
  while (fgets (line, MAXLINE, f) != NULL) {
    kwd = strtok (line, " ");
    locs[nlocs][1] = atof (kwd);
    kwd = strtok ((char *) NULL, " ");
    locs[nlocs][0] = atof (kwd);
    nlocs += 1;
    if (nlocs >= MAXLOCS) {
      fprintf (stderr, "Not enough space to hold all locations: %d!\n", nlocs);
      exit (EXIT_FAILURE);
    }
  }
  fclose (f);
  return nlocs;
}


void
coverage (float locs[][2], int nlocs, float *result, int *npaths, int *ncoin)
{
  int np_overall = 0, nc = 0, np, i, j;
  float hmsum = 0.0, dsum, hm, d, y1, x1, y2, x2;

  for (i = 0; i < nlocs; i++) {
    y1 = locs[i][0];
    x1 = locs[i][1];
    np = 0;
    dsum = 0.0;
    for (j = 0; j < nlocs; j++) {
      if (i != j) {
        y2 = locs[j][0] - y1;
        x2 = locs[j][1] - x1;
        d = sqrt (y2 * y2 + x2 * x2);
        if (d <= 0.0) {
          nc += 1;
        } else {
          dsum += 1.0 / d;
          np += 1;
          np_overall += 1;
        }
```



```
      }
    }
    hm = np / dsum;
    hmsum += 1.0 / hm;
  }
  *result = nlocs / hmsum;
  *npaths = np_overall;
  *ncoin = nc;
}


float
mean_coverage1 (char *opname, char **imsets, int nimsets)
{
  int nfiles = 0, i, fno, nlocs, np, nc;
  float locs[MAXLOCS][2];
  float csum = 0.0, cov;
  char fn[MAXFN];

  for (i = 0; i < nimsets; i++) {
    for (fno = 1; fno < 7; fno++) {
      sprintf (fn, "locations/%s_%s%d.txt", opname, imsets[i], fno);
      nlocs = load_locations (fn, locs, 0);
      coverage (locs, nlocs, &cov, &np, &nc);
      // printf ("  %s %s %f %d %d\n", opname, fn, cov, np, nc);
      csum += cov;
      nfiles += 1;
    }
  }
  return csum / nfiles;
}



int
main (int argc, char **argv)
{
  char *opnames[] = {"ebr", "haraff", "harlap", "hesaff", "heslap", "ibr",
                     "mser", "salient", "sfop", "sift", "surf"};
  int nopnames = 11;
  char *imsets[] = {"bark", "bikes", "boat", "graf", "leuv", "trees", "ubc",
                    "wall"};
  int nimsets = 8;
  int i1, i2, i3, i4;
  float mc;

  // Calculate the coverage for each individual operator.
  for (i1 = 0; i1 < nopnames; i1++) {
    mc = mean_coverage1 (opnames[i1], imsets, nimsets);
    printf ("%s %f\n", opnames[i1], mc);
  }

  return EXIT_SUCCESS;
}

//-------------------------------------------------------------------------
// End of ccoverage.c
//-------------------------------------------------------------------------
```