create a traveling salesman app in c with 50 nodes, no explanation

```
/* Traveling Salesman – Nearest-Neighbour for 50 random nodes */
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

#define NODES   50
#define MAXX    1000
#define MAXY    1000

typedef struct { double x, y; } Point;

Point pts[NODES];
int  tour[NODES + 1];          /* last = start to close loop */

/* Euclidean distance */
static inline double dist(const Point *a, const Point *b)
{
   double dx = a->x - b->x;
   double dy = a->y - b->y;
   return sqrt(dx*dx + dy*dy);
}

/* Nearest-Neighbour heuristic */
void nearest_neighbour(void)
{
   int  visited[NODES] = {0};
   int  current = 0;          /* start at node 0 */
   tour[0] = current;
   visited[current] = 1;

   for (int i = 1; i < NODES; ++i) {
      double best = INFINITY;
      int   bestj = -1;
      for (int j = 0; j < NODES; ++j)
         if (!visited[j]) {
            double d = dist(&pts[current], &pts[j]);
            if (d < best) { best = d; bestj = j; }
         }
      current = bestj;
      visited[current] = 1;
      tour[i] = current;
   }
   tour[NODES] = tour[0];      /* close the loop */
}

/* total length of the tour */
double tour_length(void)
{
```

```
    double len = 0.0;
    for (int i = 0; i < NODES; ++i)
        len += dist(&pts[tour[i]], &pts[tour[i+1]]);
    return len;
}

int main(void)
{
    srand((unsigned)time(NULL));

    /* random points */
    for (int i = 0; i < NODES; ++i) {
        pts[i].x = rand() % MAXX;
        pts[i].y = rand() % MAXY;
    }

    nearest_neighbour();

    printf("Tour order:\n");
    for (int i = 0; i <= NODES; ++i)
        printf("%d ", tour[i]);
    printf("\n");

    printf("Total length: %.2f\n", tour_length());
    return 0;
}
```