```c
/* This is one of a pair of programs to allow you to compare Java and C programs
and how they correspond and differ. They are discussed in the lectures. This in
the C version. It is not intended to be ideal C code but to match the Java code
for comparison.
*/

#include <stdio.h>  // FILE stderr fopen fclose fprintf printf fgets
#include <stdlib.h> // exit malloc free atoi abs
#include <math.h>   // lround

/* This program analyses integer salaries entered by the user.
   It outputs each salary together with its difference from the
   mean of the salaries. There must be at least one salary.
*/

// Bubble-Sort a given array of int into ascending order.
static void sort(int *anArray, int length)
{
  // Each pass of the sort reduces unsortedLength by one.
  int unsortedLength = length;
  int changedOnThisPass;
  do
  {
    int pairLeftIndex;
    changedOnThisPass = 0;
    for (pairLeftIndex = 0;
         pairLeftIndex < unsortedLength - 1; pairLeftIndex++)
    {
      if (anArray[pairLeftIndex] > anArray[pairLeftIndex + 1])
         {
           int thatWasAtPairLeftIndex = anArray[pairLeftIndex];
           anArray[pairLeftIndex] = anArray[pairLeftIndex + 1];
           anArray[pairLeftIndex + 1] = thatWasAtPairLeftIndex;
           changedOnThisPass = 1;
         } // if
    } // for
    unsortedLength--;
  } while (changedOnThisPass);
} // sort

int main(int argc, char *argv[])
{
  char *filename = argv[1];
  int index;

  // A stream for getting data from the user.
  FILE *stream= fopen(filename, "r");
  if (!stream)
  {
    fprintf(stderr, "can't open %s for reading\n", filename);
    exit(-1);
  }

  // The number of Salaries.
  int numberOfSalaries;
  // The salaries: only indices 0 to numberOfSalaries - 1 are used.
  int *salaries;

  // Initial size of the array.
  int INITIAL_SIZE = 2;
  // When the array is full, we extend it by this factor.
  int RESIZE_FACTOR = 2;

  // Read salary data from the salariesScanner,
  // count them using numberOfSalaries,
  // and store in array, extending as required.
  int salaries_length= INITIAL_SIZE;
  salaries = (int*) malloc (sizeof(int) * salaries_length);
  numberOfSalaries = 0;

#define buffer_size 100
  char data[buffer_size];
```

```
  while (fgets(data, buffer_size, stream)) //salariesScanner.hasNextLine())
  {
    // Obtain the next salary
    int currentsalary = atoi(data);

    // Extend the array if it is too small.
    if (numberOfSalaries == salaries_length)
    {
      int *biggerArray
        = (int*) malloc (sizeof(int) * salaries_length * RESIZE_FACTOR);
      for (index = 0; index < salaries_length; index++)
      {
         biggerArray[index] = salaries[index];
      }
      salaries_length*= RESIZE_FACTOR;
      free (salaries);
      salaries = biggerArray;
    } // if
    // Finally store the salary and update numberOfSalaries.
    salaries[numberOfSalaries] = currentsalary;
    numberOfSalaries++;
  } // while

  // Now compute the sum of the salaries.
  int sumOfSalaries = 0;
  for (index = 0; index < numberOfSalaries; index++)
  {
     sumOfSalaries += salaries[index];
  }

  // Compute the mean, which is a double, not an integer.
  double meanSalary = sumOfSalaries / (double) numberOfSalaries;

  // But we also want to round it to simplify the results.
  int meanSalaryRounded = lround(meanSalary);

  // Sort the salaries into ascending order.
  sort(salaries, numberOfSalaries);

  // Produce the results.
  printf("\n"
         "The mean salary is:\t%f\n"
         "which rounds to:\t%d\n"
         "\n", meanSalary, meanSalaryRounded);
  for (index = 0; index < numberOfSalaries; index++)
     // This is an alternative if-then-else in C. It is "condition ? first-option :
second-option". See if you can understand it!
  {
    int differenceFromMean = salaries[index] - meanSalaryRounded;
    char *comparisonToMean = differenceFromMean == 0.0
      ? "zero difference from"
      : (differenceFromMean < 0
         ? "less than" : "greater than");
    printf(
        "Person %2d earns %5d, which is %5d %s the mean\n",
        (index + 1), salaries[index],
        abs(differenceFromMean), comparisonToMean);
  } // for

  fclose (stream);

} // main
```