Write a C program that will calculate the gross pay of a set of employees utilizing pointers instead of array references.   Your program should continue to use all the features from past homeworks (functions, structures, constants, strings).

The program should prompt the user to enter the number of hours each employee worked. When prompted, key in the hours shown below.

The program determines the overtime hours (anything over 40 hours), the gross pay and then outputs a table in the following format.  
  
Column alignment, leading zeros in Clock#, and zero suppression in float fields is important.  
  
Use 1.5 as the overtime pay factor.

This week, adding a Total and Average row is no longer optional, its required for this assignment:

a) Add a **Total**row at the end to sum up the wage, hours, ot, and gross columns  
b) Add an **Average**row to print out the average of the wage, hours, ot, and gross columns

    ---------------------------------------------------------

    Name            Clock#   Wage   Hours     OT     Gross

    ---------------------------------------------------------

    Connie Cobol    098401   10.60   51.0    11.0    598.90

    Mary Apl        526488    9.75   42.5     2.5    426.56

    Frank Fortran   765349   10.50   37.0     0.0    388.50

    Jeff Ada        034645   12.25   45.0     5.0    581.88

    Anton Pascal    127615   10.00   40.0     0.0    400.00

    ---------------------------------------------------------

**Total:**                   53.10  215.5    18.5   2395.84

**Average:**                 10.62   43.1     3.7    479.17

You should implement this program using a structure similar to the suggested one below to store the information for each employee. Feel free to tweak it if you wish. For example, its OK to have a first and last name member instead of just a name member, and if you want to use different types, that is OK as well.

    struct employee

    {

      char  name [20];

      long  id\_number;

      float wage;

      float hours;

      float overtime;

      float gross;

    };

Set a pointer to it and then use that pointer going forward to access elements (and their associated members) in your array of structures. Again, do not use array references with indexes (use emp\_ptr->hours ... not ... emp [ i ].hours as the latter is not a fast).

Use the following information to initialize your data.

    Connie Cobol     98401   10.60

    Mary Apl        526488    9.75

    Frank Fortran   765349   10.50

    Jeff Ada         34645   12.25

    Anton Pascal    127615   10.00

Create an array of structures with 5 elements, each being of type struct employee.  
  
Initialize the array with the data provided and reference the elements of the array with the appropriate subscripts.

**Do not** use any array references with indexes. For example:

emp[i].wage /\* this is **bad**, it uses an array reference with an index, in this case, i \*/  
  
emp\_ptr->wage; /\* this is **good**, it uses a pointer to reference the wage value \*/