Case Studies

1. Range of Numbers

Problem: A survey of the computer market shows that personal computers are sold at varying costs by the vendors. The following is the list of costs (in hundreds) quoted by some vendors:

35.00,	40.50,	25.00,	31.25,	68.15,
47.00,	26.65,	29.00	53.45,	62.50

Determine the average cost and the range of values.

Problem analysis: Range is one of the measures of dispersion used in statistical analysis of a series of values. The range of any series is the difference between the highest and the lowest values in the series. That is

Range = highest value – lowest value

It is therefore necessary to find the highest and the lowest values in the series.

Output

```
Enter numbers in a line : input a NEGATIVE number to end
35 40.50 25 31.25 68.15 47 26.65 29 53.45 62.50 -1
Total values  : 10
Highest-value : 68.150002
Lowest-value  : 25.000000
Range : 43.150002
Average  : 41.849998
```

Solution:

```
Program
              main()
              {
                 int count;
                 float value, high, low, sum, average, range;
                 sum = 0;
                 count = 0;
                 printf("Enter numbers in a line :
                    input a NEGATIVE number to end\n");
input:
              scanf("%f", &value);
              if (value < 0) goto output;
                 count = count + 1;
              if (count == 1)
                 high = low = value;
              else if (value > high)
                    high = value;
                 else if (value < low)
                       low = value;
              sum = sum + value;
              goto input;
Output:
              average = sum/count;
              range = high - low;
              printf("\n\n");
              printf("Total values : %d\n", count);
              printf("Highest-value: %f\nLowest-value : %f\n",
                       high, low);
              printf("Range         : %f\nAverage : %f\n",
                       range, average);
              }
```

When the value is read the first time, it is assigned to two buckets, **high** and **low**, through the statement

```
high = low = value;
```

For subsequent values, the value read is compared with high; if it is larger, the value is assigned to high. Otherwise, the value is compared with low; if it is smaller, the value is assigned to low. Note that at a given point, the buckets high and low hold the highest and the lowest values read so far.

The values are read in an input loop created by the **goto** input; statement. The control is transferred out of the loop by inputting a negative number. This is caused by the statement

```
if (value < 0) goto output;
```

Note that this program can be written without using **goto** statements. Try.

```
#include <stdio.h>

int main() {

   int count;
   float value, high, low, sum, average, range;
   printf("Enter number in a line: Input a Negative number to end\n");

input:
      scanf("%f",&value);
      if(value < 0)
         goto output;
      count = count + 1;
      if (count == 1)
         high = low = value;
      else if (value > high)
         high = value;
         else if (value < low)
            low = value;
      sum = sum + value;
      goto input;

output:
   average = sum/count;
   range = high - low;
   printf("\n\n");
   printf("Total values : %d\n", count);
   printf("Highest-value: %f\nLowest-value: %f\n", high,low);
   printf("Range : %f\nAverage : %f\n", range,average);

   return 0;
}
```