Introduction to Programming  (Hmw 2)

Histogram is a representation of distribution of data. Distribution is given as the number of items in each bin where the bins are given as a set. For example, let us assume that the grades for an exam are the items for which the histogram/distribution is needed. And assume that the bins are the ranges given as {0-9, 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-100}. The ranges are consecutive. Histogram values would be the numbers of students in each bin; that is, the number of students that have 0-9, the number of students that have 10-19, etc.

Your job is to write a function that calculates histogram values (counts). The prototype of your function shall be

```c
void Histogram(int Data[], int N, int Bins[], int M, int Counts[]);
```

where

- `Data[]`: input data given in an array of integers (grades for example)
- `N`: the number of students/grades
- `Bins[]`: low end of the ranges of the bins. For example, for the 0-9 range it is 0.
- `M`: the number of bins (10 for the example given above)
- `Counts[]`: Number of students in each bin. For example `Counts[0]` will contain the number students that received a grade between 0 and 9 inclusive. Size of `Counts[]` is `M` too.

Test your function using 120 grades (`Data[]`) in the range 0-100 that are generated randomly in function `main()`. Initialize your `Bins[]` according to the example given above. Declare all your arrays in `main()`. Display the results (`Counts[]`) in `main()`.

Hint: For a uniform random data, all bins are expected to have counts close to others.

**Answer:**

```c
#include <stdio.h>
#include <stdlib.h>

void Histogram(int Data[], int N, int Bins[], int M, int Counts[]){
  int i, k;
  for (i=0; i<M; i++){
    Counts[i]=0;
    for (k=0; k<N; k++)
      if ((Data[k]>=Bins[i]) && (((i==M-1) || (Data[k]<Bins[i+1]))))
        Counts[i]+=1;
  }
}

int main(void){  /* try the function using random data */
  int i, D[120], C[10], B[10] = {0, 10, 20, 30, 40, 50, 60, 70, 80, 90};
  for (i=0; i<120; i++) D[i] = (int) (100.99 * rand() / RAND_MAX);
  Histogram(D, 120, B, 10, C);
  for (i=0; i<9; i++)
    printf("%2d-%2d : %d\n", B[i], B[i+1]-1, C[i]);
  printf("%2d- : %d\n", B[9], C[9]);
  return 0;
}
```