How to debug a program

General Rules

- Read your program again and think about the logic.
- Make your own data sets and check the answers according to a baseline program.
  - Of course, you need a baseline program in begin. It can be simply a basic search program.
- If you have a debugger (e.g. GDB, Visual Studio), set breakpoints and see if the values of variables (both local and global ones) match what are expected.

POJ 2559 - **Largest Rectangle in a Histogram**

- Description
- …

Input

The input contains several test cases. Each test case describes a histogram and starts with an integer n, denoting the number of rectangles it is composed of. You may assume that $1<=n<=100000$. Then follow n integers $h_1,...,h_n$, where $0<=h_i<=1000000000$. These numbers denote the heights of the rectangles of the histogram in left-to-right order. The width of each rectangle is 1. A zero follows the input for the last test case.

Output

For each test case output on a single line the area of the largest rectangle in the specified histogram. Remember that this rectangle must be aligned at the common base line.

- Sample
- …

How to make a proper data set?

- Follow the input description strictly.
  - “The input contain several test cases.”
    - nTest := 100 (or other values you like)
  - “…You may assume n<=100000”
    - n := random()%100000+1
  - Here random() denotes a random integer generator.
  - “Following n integers, … 0 <= h_i <= 1000000000”
    - Foreach x in h: x:=random()%1000000001
  - “A zero follows … the last test case.”

How to make a proper data set?

- The C++ program making the data set can be written like the following:
  - …
  - int random()// return a random variable between 0..2^31-1
  - …
  - int main(){
    - int ntest = 100;
    - while(ntest--){
      - int n = random()%100000+1; printf(“%d\n”,n);
      - for(int i=0; i<n; i++)printf(“%d “,random()%1000000001); puts(“”);
    - }
    - printf(“0\n”);
  - }

How to make a proper data set?

- Redirect the output file to a local .txt(.log, etc) file for further use. In latter case I’ll use “input.txt” to denote the generated input file.

Baseline Program

- To the previous program, an O(n^2) algorithm is very easy:
  - For each rectangle, calculate the number of continuous rectangles with height >= current height in left and eight in left and right by scanning the array directly.
- Not all problem have a baseline program as simple as above.
- Consider the ability of this baseline program and modify the parameters in the data making program to make a proper data set.
  - For the previous program, an O(n^2) program can solve cases with n<=10000 within time.
  - So in data making program n should be random()%10000+1.

Checking Program

- Now you can use your program to generate an output file (“output.txt” in latter case), then use the baseline program to generate an output file (“std.txt” in latter case).
- In Windows 8, use command “fc output.txt std.txt” can check if the two output file matches.
- You may also use Vim command “vimdiff output.txt std.txt”
- Other operating systems also have file comparing commands.

Debugging

- If you don’t find any difference:
  - Re-make a data set and check again.
- If some difference found:
  - Find the corresponding input in “input.txt”
  - Use a debugger to check if your program works properly by setting breakpoints and stepping the program