Sorting by combining merge and insertion

Description We will combine two sorting algorithms, insertion-sort and merge-sort as follows. When the input size is less than 10. Otherwise, we use Merge-sort. More specifically, we replace line 1 in Merge-Sort (page 34) with "if $r-p >= 10$", and add line 6 "else Insertion-Sort(A, p, r)". Insertion-Sort(A, p, r) implies performing insertion sort on the subarray $A[p \cdots r]$). Implement this hybrid sorting algorithm.

For the test of your codes, copy "Grade01" to your working directory. You also need to copy the directory "testfiles" in your working directory along with all files under the "testfiles" directory: sol.exe, t1, t2, ..., t10. All files are archived in "Lab01.tar" which can be found in the files tab in catcourse. Your execution file's name must be "MergeInsertion.exe". Perhaps you may need to use chmod to change the permissions of "Grade01" and "sol.exe" under the "testfiles" directory. Towards this end, you can just type

```
chmod 700 Grade01
chmod 700 ./testfiles/sol.exe
```

Now if you run "./Grade01" and your implementation is correct, you will see the following messages.

```
rm: result: No such file or directory (you may or may not see this)
rm: tempresult1: No such file or directory
rm: tempresult2: No such file or directory
Correct for 1 th example.
Correct for 2 th example.
Correct for 3 th example.
Correct for 4 th example.
Correct for 5 th example.
Correct for 6 th example.
Correct for 7 th example.
Correct for 8 th example.
Correct for 9 th example.
Correct for 10 th example.
```

You can find the log of the execution in the file named "result".

You are required to compile your codes in front of the TA, and answer any questions you are asked about your codes. The idea is that you should understand what your wrote. If you want to test your code for just one test file, you can try: ./MergeInsertion.exe < ../testfiles/t1

Finally, I also wrote a very simple "Makefile" for you, which is a very convenient tool for compiling multiple files together. If you type "make," it will compile MergeInsertion.cpp (sorry, if you have other files, this won't work. You have to modify Makefile) If you type "make clean", then MergeInsertion.exe will be deleted. You don't have to use "Makefile". But if you don't know how to use it, perhaps it is a good time to learn it. You can find abundant examples on the web.

Input structure The input starts with an integer number which indicates the number of elements (integers) to be sorted. Then, the elements follow, one per line.

Output structure Output the sorted sequence one element per line. Do *not* insert spaces at the beginning or at the end of any element.

Examples of input and output

Example:
Input

```
6
3
6
23
76
4
56
```

Output

```
3
4
6
23
56
76
```

Your solutions Before leaving the lab, submit a zipped tar archive of your program through the assignments page of CatCourse. Please use your UCMNetID as the filename for the zipped tar archive. Be careful since CatCourse strictly enforces the assignment deadlines (deadlines will be every lab date at 10:30am or 1:30pm depending on your lab session).