

Introduction to Data Structures

Jinkyu Lee

Dept. of Computer Science and Engineering,
Sungkyunkwan University (SKKU)

Homework 1A

- 40 points for coding evaluation
 - Submission format
 - File name: yourid_HW1A.c
 - Example: 2000123456_HW1A.c
 - File type: .c (NOT .cpp)
 - Submission site: <https://icampus.skku.edu>
 - Week 2: [Homework] 1A (code)

- 1 point for report
 - The report is not evaluated in detail but evaluated as Pass/Fail
 - Template: Homework Report Template.docx
 - Submission format
 - File name: yourid_HW1A.pdf
 - Example: 2000123456_HW1A.pdf
 - File type: .pdf
 - Submission site: <https://icampus.skku.edu>
 - Week 2: [Homework] 1A (report)

- Due date
 - 9/22 23:59 -> 9/27 23:59 (no late submission accepted)

Rules for homework

- You should follow instructions.
 - Compiler
 - You will get **no/less point** if your program cannot be complied with the specified compiler
 - Input/output format
 - You will get **no/less point** if TA's automatic evaluation program cannot parse your input or output.
 - Permitted modification scope
 - You will get **no/less point** if you modify code outside of the permitted modification scope
 - All other rules
 - You will get **severe penalty or no/less point** if you violate the given rules.

Compiler and input/output rules for homework

- Every implementation homework will be evaluated by TA's automatic evaluation program with the following compiler.
 - Compiler: GCC 7.X, 8.X, 9.X or 10.X
 - <https://gcc.gnu.org/>
 - You will get no/less point if your program cannot be compiled with GCC 7.X, 8.X, 9.X or 10.X.
 - For example, do not rely on visual studio.
 - You can use standard library such as *stdlib.h* and *math.h*.
- Input/output format
 - You will get no/less point if TA's automatic evaluation program cannot parse your input or output according to the following rules.
 - Use `stdin` and `stdout`

Problem

■ Problem: 4-ary number

- In a 4-ary number, each digit has the number 0, 1, 2, or 3.
- Given the number of digits N (1~6) and the sum of all digits X (0~18), print all the 4-ary number that satisfies the following conditions.
 - The number of digit is N ; and
 - The sum of all digits are equal to ~~or larger than~~ X .
- You should print the numbers in an ascending order.
- You should use **iteration** (You should **not use recursion**).
- Each line of output should have N -digit numbers.
 - For example, if $N=3$, you should print “001” instead of “1”.
- The order of printed numbers should be sorted in ascending order.
- If there is no number that satisfies the condition, you should print nothing.

Input/Output

■ Sample1 (N=2, X=3)

input	output
2	03
3	12
	21
	30

■ Sample 2 (N=4, X=10)

input	output
4	1333
10	2233
	2323
	2332
	3133
	3223
	3232
	3313
	3322
	3331

Evaluation

■ Evaluation

- TA will test several cases.
- For each test case,
 - If your C code does not have any recursive function and the result is printed within 10 seconds on a platform with average computing power,
 - If your output is perfect in terms of the values and the order,
 - You get 100%.
 - Else if your output is perfect in terms of the values only (not the order),
 - You get 70%.
 - Else if your output contains any wrong values,
 - You get 0%.
 - Else
 - You get $70\% * (\# \text{ of your values}) / (\# \text{ of the values for the perfect answer})$.
 - Else,
 - You get 0%.