60-141 Introduction to Programming

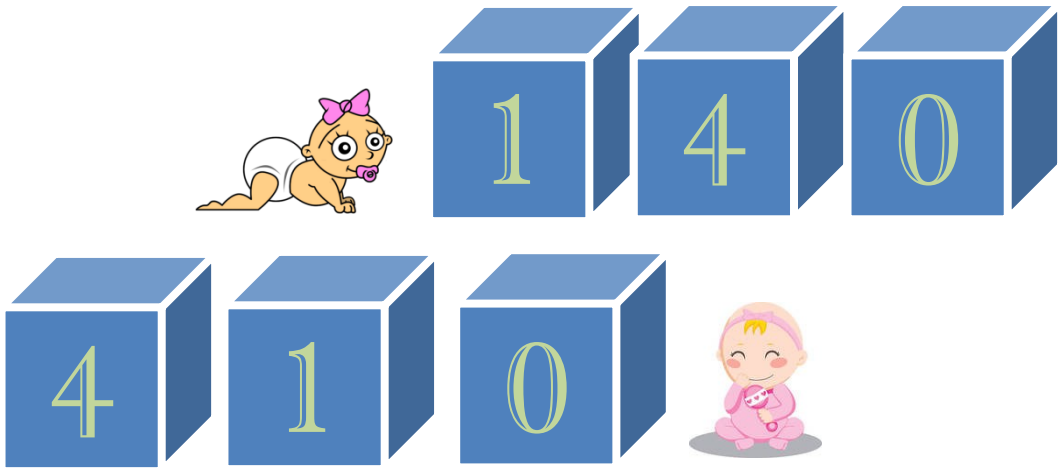
Assignment 1: A Simple Children's Game

(modified from the UVA online judge problem set)

Due date: Friday October 10, 11:59pm

Instructor: Dr. Ziad Kobti

There are lots of number games for children. These games are pretty easy to play but not so easy to make. We will discuss about an interesting game here.



Baby G's mom discovered her daughter's apt to algorithms and problem solving soon after her second birthday. Baby G was playing with number blocks, each labelled with a digit from 0 to 9.
Mom noticed that Baby G had the ability to always create a combination of blocks that would form the largest number. For example: if she had the blocks 0, 1, and 4, she would align them from left to right as 4,1,0. Baby G knew that the 3 blocks will form the number 410, which is larger than 104, or 140, or 014, or 041, or even 401! Now mom needs your help to authenticate Baby G's talent, especially since more blocks are involved! In fact, Baby G loves playing blocks so much that her mom brought her even more sets… You may think that it's very easy to find out the answer but will it be easy for a child who has just got the idea of number?

You need to write a program that can input 5 digits in any order, and output the largest integer that can be composed of these digits!

Sample input:
`1 4 5 3 2`
Sample output:
`54321`

Another sample input:
`9 8 2 9 9` (remember, she has now more block sets so could have multiple blocks of any given number!)
Sample output:
`99982`

What you are asked to do:

1. Write a complete algorithm explaining your logic in the form of a pseudocode /Flowchart using **Raptor** (assign1.rap). **[5 points]**
2. Write a complete C program, fully documented, showing how you can read the digits and display the largest possible number. **[10 points]**
3. Create the script file clearly testing your code with different test cases and submit the assignments before the due date using the CLEW. [2 points]

Preparing your C code for submission:

Call your program assign1.c
Create a script file as follows:
0. cp assign1.c assign1.c.bak [backup your file by creating a copy under different name!]
1. script assign1.txt [careful! if you type assign1.c instead you erase your code!]
2. cat assign1.c
3. cc assign1.c
4. a.out [test your program thoroughly, run it a few times and test it with at least 3 different test cases]
5. ls -l
6. exit [do not forget this step!!!]

Submit the files: assign1.txt, assign1.c, and assign1.rap in the CLEW/Assignment menu for the course.

NOTES:
1. Your assignment must be RECEIVED by the due date and time. Late assignment submissions are NOT accepted. Keep your script file, and all your code unmodified as proof of its completion in case it is not received.
2. It is your responsibility to get an early start on the assignment, research and ask questions ahead of time from the due date.

3. You must use your own uwindsor account to submit your work on CLEW.
4. Undocumented or improperly documented code will NOT be graded and will receive a mark of ZERO.
5. Marks will be deducted for unclear code. (improper spacing and alignment, hard to read programs and missing outputs).
6. Make sure you turn in a complete script file that clearly shows: your code, your compilation process, a listing of the directory showing your source file(s) and the a.out with the date/time stamps, and the output.
7. **PLAGIARISM:** CHEATING IS NOT TOLERATED. You must submit your own work. Students who are suspected of copying someone else's work will be reported to the department's chair and the Dean of Science and be dealt with in accordance with the University policies. You should not share your code with others. Codes that are similar to each other will BOTH be reported as evidence of copying. You are strongly encouraged to write your own code.
8. Authorized/limited help on this assignment may be provided directly from your Lecture or Lab instructors and Teaching Assistants.