Institut for Matematik og Datalogi
Syddansk Universitet

Assignment 4 — Introduction to Computer Science 2013–14

This is your fourth assignment in DM534. The assignment is due at 8:15 on Thursday, February 20. You may write this either in Danish or English. It must be made in LaTeX. (though you do not need to include your LaTeX code). Write your full name, your section number, and your "instruktor"s name (Magnus Gausdal Find or Christian Kudahl — it is not yet certain who will have S7 and who will have S17, but this will be decided by the beginning of February) clearly on the first page of your assignment (on the top, if it's not a cover page). Turn in an electronic version as a PDF file via Blackboard through your DM534 course (choose the correct one, S7 or S17). The assignment hand-in is in the menu for the course and is called "SDU Assignment". Keep the receipt it gives you proving that you turned your assignment in on time. Blackboard will not allow you to turn in an assignment late.

Cheating on this assignment is viewed as cheating on an exam. You are allowed to talk about course material with your fellow students, but working together on this assignment is cheating. If you have questions about the assignment, come to Joan Boyar or your "instruktor" for DM534.

Please note that you must have this assignment approved in order to pass DM534. If it is not turned in on time, or if you do not get it approved, it will count as one of your two retries (assuming you still have at least one retry remaining) in the course, and you must have it approved on your only allowed retry for this assignment.

Assignment 4

Do the following problems and write your solutions in LaTeX. Write clear, complete answers, but not longer than necessary. Do not include the statements of the problems or other information not asked for in the problems.

1. Do one of the following two programming problems in Bare Bones. Remember to have comments in your program. (Put comments between curly brackets: {, }.) If you use a command not in Bare Bones, such as the COPY command defined in the textbook, define (write) the subroutine (piece of code) for that command. The second problem is the more challenging.

   (a) Write a Bare Bones program to compute the value $3u + 2v$, where u is the value in the variable U and v the value in V. At the end of the execution of your program, the value $3u + 2v$ should be in a variable Z. It is OK if the values in the variables U and V change.

   (b) Write a Bare Bones program to calculate the sum of values $1 + 3 + 5 + ... + x$, where x is an odd positive integer value in the variable X. At the end of the execution of your program, the sum of the first odd integers, up to and including x should be in a variable Z. It is OK if the value in the variable X changes.

2. Construct one of the following two Turing machines. In both cases, assume that the Turing machine tape is infinite in both directions, that it has some input initially on some consecutive cells, and that all other cells have asterisks (*). Assume that the read/write head is initially placed on the first asterisk to the right of the cells with other symbols. You may assume that there is at least one cell without an asterisk. When your Turing machine halts, the required output should be in consecutive cells, and all other cells should have asterisks.

   Express your Turing machine using a table, as in Figure 12.3 in the textbook. Choose your own (meaningful) names for states. Use as many as you want. Explain what your Turing machine does, also.

   Note that there is a TM simulator on the course's homepage. Beware that you need to place * symbols on the locations where your TM might actually expect to find them. The program doesn't automatically put some extras in. Then you can test your Turing machine on interesting inputs, but do not include these tests in your answer to this problem.

   Again the second problem is the more challenging.

   (a) Suppose the input string on the Turing machine's tape, is an integer in base 10, i.e., each cell contains a digit

$$d_i \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}.$$

Design a Turing machine which starts on the asterisk just to the right of the input and adds 37 to this number. Note that this might involve moving an asterisk to the left (changing an asterisk to a digit), and that the the original number could be larger than or smaller than 10.

(b) Design a Turing machine that has as input a string of zeros and ones, considers the string as representing a non-negative integer in binary form in the usual way, and multiplies it by 10 (ten). As in the other problem, this may involve changing some asterisks to something else.