

cis1.5-spring2007-sklar, lab V, part 2

instructions

- This is the second part of the lab/homework assignment for unit V.
- The entire assignment will be worth 9 points.
- The first part is worth 6 points and was distributed and worked on in class on Wednesday April 25 and Wednesday May 2.
- The second part is worth 3 points and will be distributed in class on Wednesday May 2.
- **Both parts together are due on Monday May 7** and must be submitted by email (as below). NOTE that part 2 may be submitted as hardcopy in class on May 7.
- **Follow these emailing instructions:**
 1. Create a mail message addressed to `sklar@sci.brooklyn.cuny.edu` with the subject line **cis1.5 hw5**.
 2. Attach ONLY the **.cpp** files for each part, as outlined below.
DO NOT ATTACH THE **.cbp** (CodeBlocks Project) files!
 3. Failure to follow these instructions will result in points being taken away from your grade. The number of points will be in proportion to the extent to which you did not follow instructions... (which can make it a lot harder for me to grade your work — grrrr!)

sorting.

For this assignment, refer to the examples we discussed in class on Monday April 30.

You do NOT need to submit any programs for this part of the assignment! Create your answers in a text file and email them to me along with the first part, or bring a hardcopy to class on May 7.

In class on Monday April 30, we discussed several *sorting* algorithms. I provided some examples and posted these on the class web page: `selectionsort.cpp`, `insertionsort.cpp` and `bubblesort.cpp`. (Ignore `blortsort.cpp` for this assignment.)

For each of these three algorithms (1 point each), do the following:

1. Assume that the dice array is initialized to:

3	6	1	6	5
---	---	---	---	---
2. After each pass of the algorithm, determine the order of the entries in the dice array.

You must submit, for each of the three sorting algorithms (selection sort, insertion sort and bubble sort), the number of passes it takes to complete the sort and, using the above array as the initial setting, the contents of the array after each pass.

Hint: Do it on paper first. Then check your answers by downloading the code from the class web page, modifying `initDice()` to initialize using the values listed above, and run each algorithm.