Introduction to Programming, PIC10A
E. Ryu
Spring 2017

UCLA

Homework 5
Due 5pm, Wednesday, May 10, 2017

We may take off up to 20% of the total marks for poor style; make sure to name your variables reasonably, indent properly, and comment sufficiently. Submit `sort.cpp`.

Problem 1: (Sorting)

Write a program that sorts `n` integers, from smallest to largest, with what's called the *selection sort* method. The input and output should be exactly

```
How many numbers do you have?
[USER ENTERS A POSITIVE INTEGER]
Input number (count 1):
[USER ENTERS A INTEGER]
Input number (count 2):
[USER ENTERS A INTEGER]
...
Input number (count X):
[USER ENTERS A INTEGER]
The sorted input is:
X (count 1)
X (count 2)
...
X (count X)
```

You may not use any libraries aside from `iostream` and `string`.

Hint. There are many ways to sort an array, and some are quite sophisticated. In my opinion, the selection sort method is the simplest (although not the best). We suggest you write your program in the following steps.

- Get the number `n` from the user.
- Create a dynamically allocated array of size `n`.
- Store the numbers in the array.
- Find the smallest entry among entries `0` to `n-1`.
- Swap the smallest entry with entry `0`. I.e., store the value of the `0`th `int`, write the smallest `int` into the `0`th spot, and write the stored `int` into the spot the smallest `int` used to occupy.
- Find the smallest entry among entries `1` to `n-1`, and swap the smallest entry with entry `1`.

- Find the smallest entry among entries `2` to `n-1`, and swap the smallest entry with entry `2`.
- ...
- Find the smallest entry among entries `n-1` to `n-1` (which is simply entry `n-1`), and swap the smallest entry with entry `n-1` (so do nothing).
- Output the array.
- Deallocate the array.

The swapping process is illustrated well in:
`https://en.wikipedia.org/wiki/Selection_sort#/media/File:Selection-Sort-Animation.gif`

Hint. In the swapping process, you'll have to keep track of the index of the smallest `int`. The following pseudocode illustrates how to find the index of the smallest `int` among entries `0` to `n-1`.

```
min_ind = 0
for (i=1,2,...,n-1)
  if (array[i] < array[min_ind])
    min_ind = i
```