CS311
Kazumi Slott
Homework on T(n) of selection sort

Type or handwrite your work and answer. Please draw a box around your final answer. You will receive 0, if it is not legible. It is due at the beginning of lecture. No submissions will be accepted after lecture starts. If you need to skip class or arrive late, you need to email me your file BEFORE lecture starts (-3 points charged for printing).

<Question> Count the number of comparisons (if) needed to sort N elements in the following selection sort algorithm. Express your answer as a function of T. T(N) = ???????????

```
//this function sorts the numbers in ascending order by moving the largest to the end
void selectionSort1(int array[], int N)
{
  int lrgIndx; //the index of the largest value
  int temp; //temporary variable that holds the largest value

  //last is the last index in unsorted part
  for(int last = N-1; last >= 1; last--)
    {
      lrgIndx = 0; //assume the first item is the largest
      //find the largest in unsorted part ([0..last])
      for(int i = 1; i <= last; i++)
      {
       if(array[i] > array[lrgIndx]) //The current item is larger
          lrgIndx = i;
      }

      //swap the largest with the last item in the unsorted part
      temp = array[lrgIndx];
      array[lrgIndx] = array[last];
      array[last] = temp;
    }
}
```

Suggestion:
You might want to fill the following table to figure out.

last	N-1	N-2	.....
# of comparisons	?	?	

HINT:
This algorithm is explained on pages 17 and 18 (the last 2 pages) of my CS111 "Lecture Notes 6-1: arrays - single dimensional. If you don't understand this algorithm, I suggest you watch my CS111 recording "Recording of lecture on November 13(Fri) - binary search and selection sort" from 26:05.