CSCE 221 Cover Page
Programming Assignment #1
Due July 10th by midnight to eCampus

First Name Michael Last Name Mengistu UIN 125000724

July 10, 2018

User Name michaelmengistu E-mail address michaelmengistu@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more: Aggie Honor System Office

Type of sources			
People			
Web pages (provide URL)			
Printed material			
Other Sources			

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.

"On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work."

Your Name Michael Mengistu Date July 10, 2018

Program Description: Make a abstract data type that stores chars into a dynamic array and another abstract data type that stores any typename into a dynamic array that can do the following operations:

- let the user insert a new element at a given rank.
- let the user index threw the array.
- let the user return a element at a given rank without removing it.
- let the user replace a element in the array at a given rank with a new element.
- let the user remove a element at a given rank.
- resize the array if the capacity of the array is full.
- find the max index of the largest element in the array.
- sort the array the array from smallest to largest.

Data Structures Description:

- elem_at_rank(int r): returns the element at rank r. The best and worst run time is O(1).
- replace_at_rank(int r, const char& elem): replaces a element in the array at r with elem. The best run time O(1) and the worst run time is O(n).
- insert_at_rank(int r, const char& elem): insert a new element at r with elem and resize the array by double if the capacity of the array is full. The best and worst run time is O(nlogn).
- remove_at_rank(int r): remove a element at r. The best and worst run time is O(1).
- find_max_index(const My_vec& v,int size): find the max index of the largest element in the given size of array. The best run time is O(1) and worst run time is O(n).
- sort_max(My_vec& vec): sort the array from smallest to largest using the function find_max_index. The best run time is O(n) and worst run time is O(n^2).

Instructions to Compile and Run your Program; Input and Output Specifications:

- Compile your program using the Linux machine command line: g++ -std=c++11 *.cpp or make all
- Run your program by executing: ./Main
- Input and Outputs :
  - get_size(): Input- none Output- size of array
  - get_capacity(): Input- none Output- amount of elements you can store in array
  - operator[]: Input- index Output- element at index
  - is_empty(): Input- none Output- true if there is elements in array, false otherwise.
  - elem_at_rank(int r): Input- rank Output- element at index
  - replace_at_rank(int r, const char& elem): Input - rank, element Output - None
  - remove_at_rank(int r): Input- rank Output- none
  - insert_at_rank(int r, const char& elem): Input- rank, element Output- none

C++ generic programming features:

- let you make a class that can handle all nametypes

Testing results:

- when testing results my outputs I followed the comments given to me on the main.cpp file and got the right outputs that I expected.