Programming Assignment 1
CSCE 221

1. The description of an assignment problem.

   (a) The My_vec program is a C++ class that holds character data types and contains functions to manipulare this data, such as an insert, replace, and remove functions. As well as helper function such as a function to find the index of the largest index and a sort function that will sort each from largest to smallest values. From this program a template vector class was created so that multiple data types could be supported.

2. The description of data structures and algorithms used to solve the problem.

   (a) Provide definitions of data structures by using Abstract Data Types (ADTs)

      i. The data structures that this program uses is based on stack and heap memory allocation by using pointers and mimicing the structure of an array.

   (b) Write about the ADTs implementation in C++.

      i. These Abstract Data Types are implemented in the defining of each member function, specifically the inset function which relies on the array qualities as well as the structures of the stack and heap.

   (c) Describe algorithms used to solve the problem.

      i. There are three main algorithms that are used within the My_vec class:

         A. insert_at_rank: This algorithm will resize the vector arry when the size of the arry equals capacity of the array, as well as allowing the insertion of an element to any rank of the vector so long as there are no gaps between any two elements.

         B. replace_at_rank: This algorithm will find an already exsisting element and replace the value at the given rank with the given value of the same type.

         C. remove_at_rank: This algorithm will find an already esisting element and remove the value and decrement the size of the array.

   (d) Analyze the algorithms according to assignment requirements.

      i. The best and worst case scenarios for the vector algorithms are O(1) is the best case senario because the least amount of comparsions means going through the for the loop only once. O(n) is the worst case senario because it will take n-1 comparisions

3. A C++ organization and implementation of the problem solution
   (a) Provide a list and description of classes or interfaces used by a program such as classes used to implement the data structures or exceptions.
      i. class My_vec:
         A. elem_at_rank: returns the value of the element at the desired rank/index
         B. insert_at_rank: inserts element at avalible rank(index) preventing empty gaps in the vector and doubling the capacity if the vector size has reached capacity
         C. replace_at_rank: replaces the value of the element at the desired rank/index
         D. remove_at_rank: removes the value of the element at the desired rank/index and decriments the size
         E. constructors: allocates and initializes an instance of the vector class
         F. copy consturctors, assignment operator: copies one vector to another using a deep copy
         G. destructor: deallocates memory of a given vector
         H. overloading bracket [] operator: allows the access of vector index
      ii. helper functions
         A. find_max_index: finds the index of the largest value in the vector
         B. sort_max: uses the find_max_index function to sort the vector from largest to smallest value
   (b) Include in the report the class declarations from a header file (.h) and their implementation from a source file (.cpp).
      i. Part 1: My_vec
         My_vec.h
         My_vec.cpp
         main.cpp
      ii. Part 2: TemplatedMy_vec
         TemplatedMy_vec.h
         main.cpp
   (c) Provide features of the C++ programming paradigms like Inheritance or Polymorphism in case of object oriented programming, or Templates in the case of generic programming used in your implementation.
      i. Polymorphism is present in mulitply member functions where these size of the vector is needed. There is also polymorphism in the helper functions where sort_max calls find_max_index to sort the vector.

4. A user guide description how to navigate your program with the instructions how to:
   (a) compile the program: specify the directory and file names, etc.
      i. Part 1: My_vec
         A. cd my_vec
            My_vec.h
            My_vec.cpp
            main.cpp
            My_vec.o
            main.o
            Makefile
         B. make all
      ii. Part 2: my_vecTemplate
         A. cd my_vecTemplate
            TemplatedMy_vec.h
            main.cpp
            main.o
            Makefile
   (b) run the program: specify the name of an executable file.
      i. ./main

5. Specifications and description of input and output formats and files

   (a) The type of files: keyboard, text files, etc (if applicable).

      i. no input needed

   (b) A file input format: when a program requires a sequence of input items, specify the number of items per line or a line termination. Provide a sample of a required input format.

      i. no input needed

   (c) Discuss possible cases when your program could crash because of incorrect input (a wrong file name, strings instead of a number, or such cases when the program expects 10 items to read and it finds only 9.)

      i. no input needed

6. Provide types of exceptions and their purpose in your program.

   (a) logical exceptions (such as deletion of an item from an empty container, etc.).

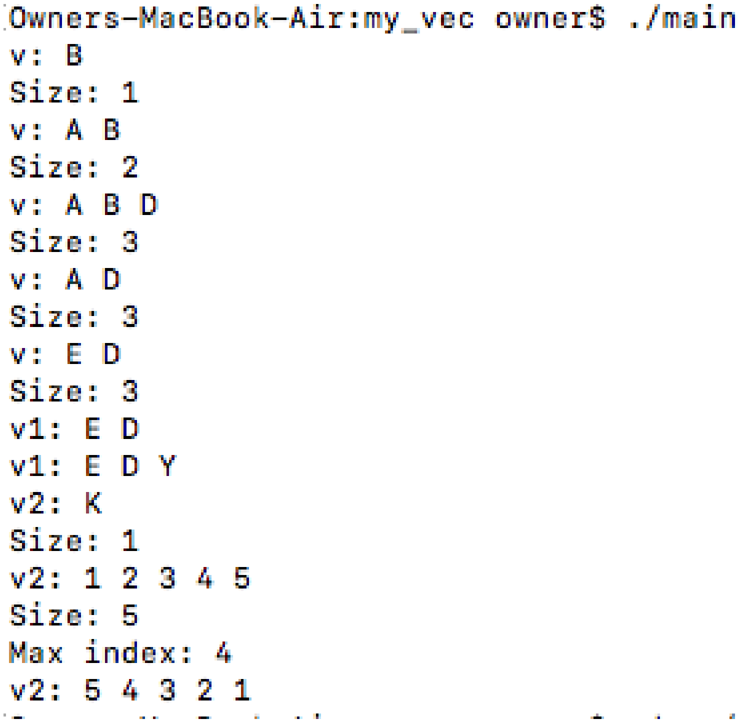
      i. Exception thrown when insert_at_rank when inserting and gaps are present in the data

   (b) runtime exception (such as division by 0, etc.)

      i. Exception thrown when insert_at_rank when inserting and gaps are present in the data

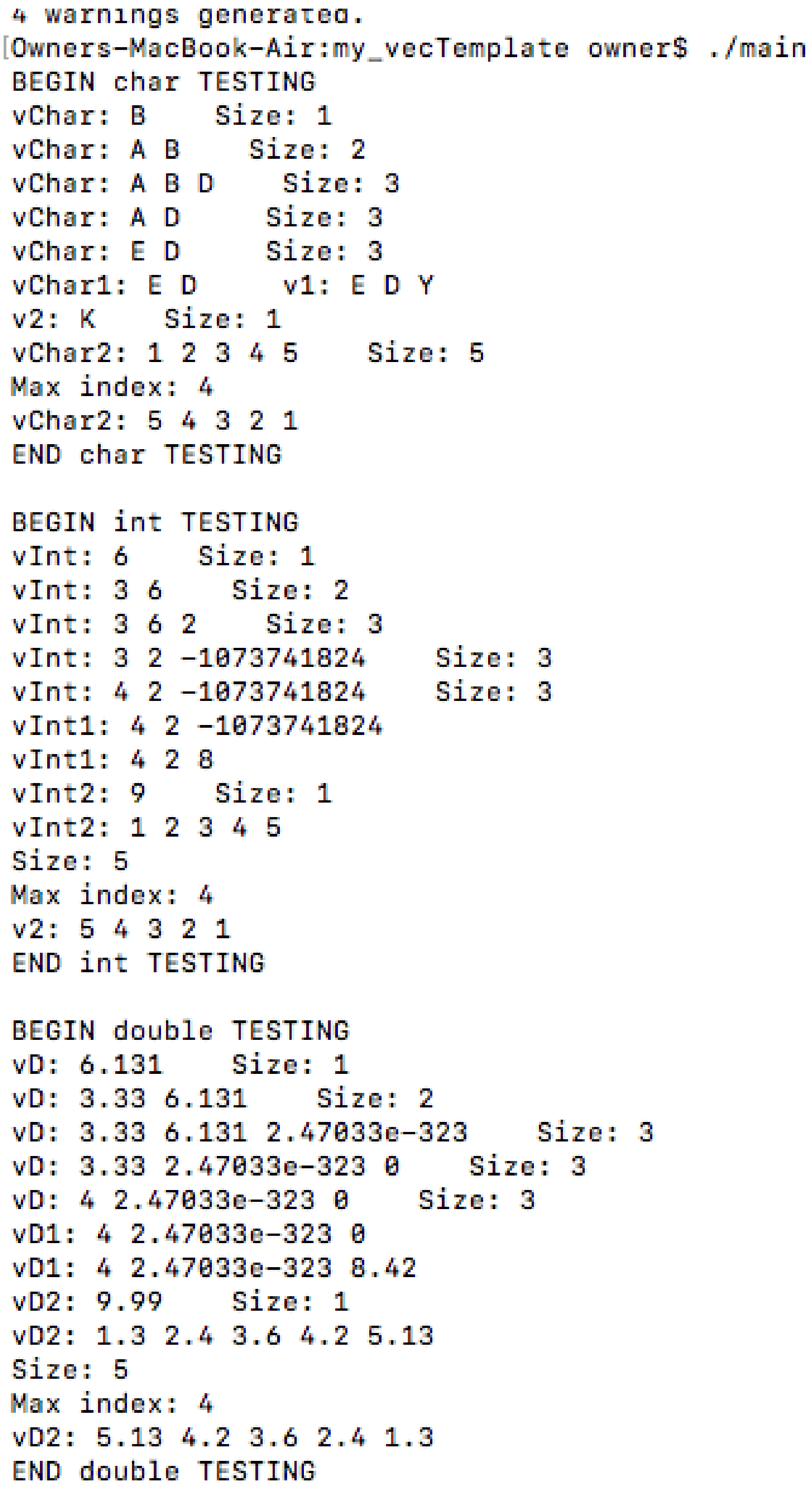
7. Test your program for correctness using valid, invalid, and random inputs (e.g., insertion of an item at the beginning, at the end, or at a random place into a sorted vector). Include evidence of your testing, such as an output file or screen shots with an input and the corresponding output.

   (a) My_vec



```
Owners-MacBook-Air:my_vec owner$ ./main
v: B
Size: 1
v: A B
Size: 2
v: A B D
Size: 3
v: A D
Size: 3
v: E D
Size: 3
v1: E D
v1: E D Y
v2: K
Size: 1
v2: 1 2 3 4 5
Size: 5
Max index: 4
v2: 5 4 3 2 1
```

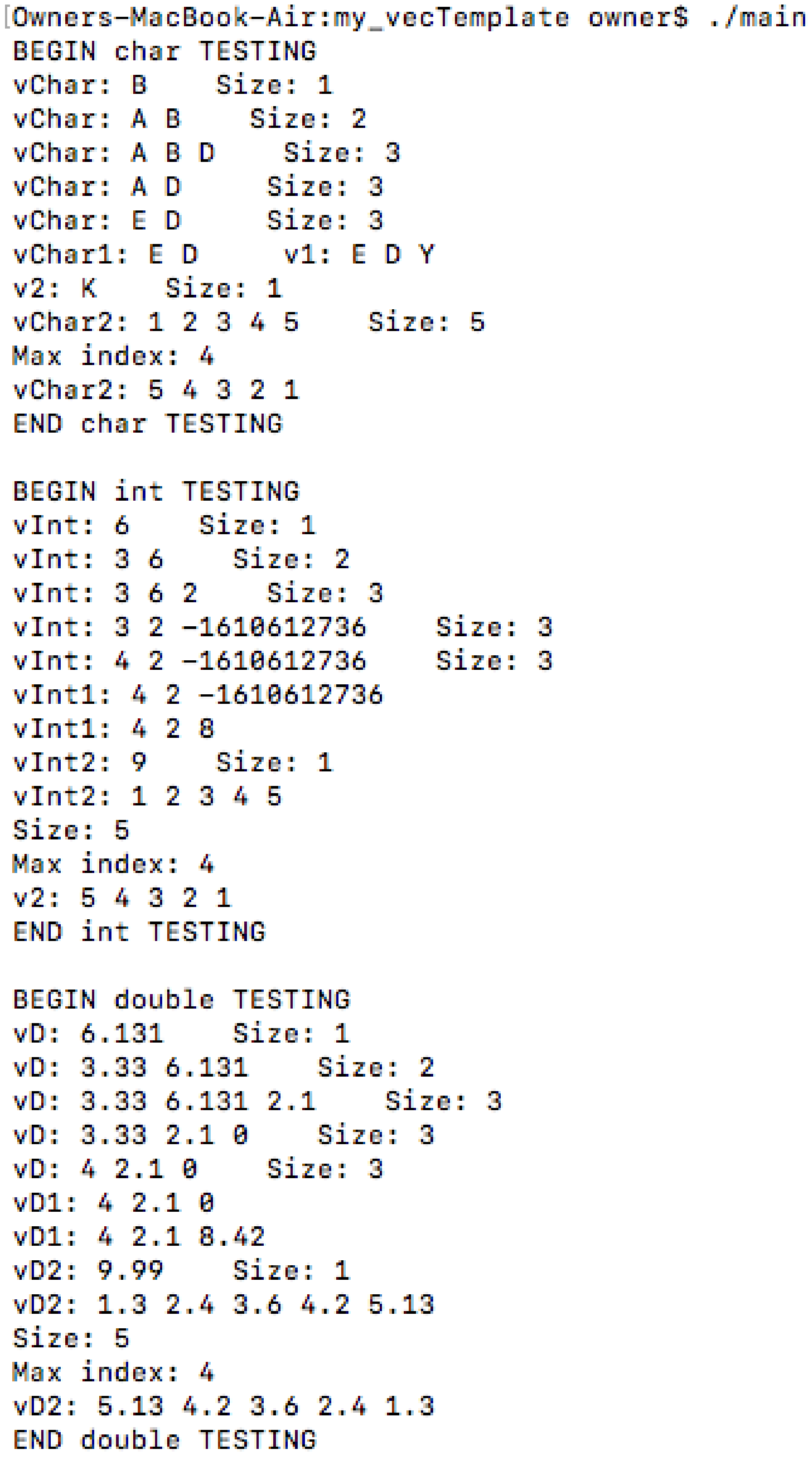
   (b) TemplatedMy_vec



```
4 warnings generated.
[Owners-MacBook-Air:my_vecTemplate owner$ ./main
BEGIN char TESTING
vChar: B    Size: 1
vChar: A B    Size: 2
vChar: A B D    Size: 3
vChar: A D     Size: 3
vChar: E D     Size: 3
vChar1: E D     v1: E D Y
v2: K    Size: 1
vChar2: 1 2 3 4 5    Size: 5
Max index: 4
vChar2: 5 4 3 2 1
END char TESTING

BEGIN int TESTING
vInt: 6    Size: 1
vInt: 3 6    Size: 2
vInt: 3 6 2    Size: 3
vInt: 3 2 -1073741824    Size: 3
vInt: 4 2 -1073741824    Size: 3
vInt1: 4 2 -1073741824
vInt1: 4 2 8
vInt2: 9    Size: 1
vInt2: 1 2 3 4 5
Size: 5
Max index: 4
v2: 5 4 3 2 1
END int TESTING

BEGIN double TESTING
vD: 6.131    Size: 1
vD: 3.33 6.131    Size: 2
vD: 3.33 6.131 2.47033e-323    Size: 3
vD: 3.33 2.47033e-323 0    Size: 3
vD: 4 2.47033e-323 0    Size: 3
vD1: 4 2.47033e-323 0
vD1: 4 2.47033e-323 8.42
vD2: 9.99    Size: 1
vD2: 1.3 2.4 3.6 4.2 5.13
Size: 5
Max index: 4
vD2: 5.13 4.2 3.6 2.4 1.3
END double TESTING
```



```
[Owners-MacBook-Air:my_vecTemplate owner$ ./main
BEGIN char TESTING
vChar: B    Size: 1
vChar: A B    Size: 2
vChar: A B D    Size: 3
vChar: A D     Size: 3
vChar: E D     Size: 3
vChar1: E D     v1: E D Y
v2: K    Size: 1
vChar2: 1 2 3 4 5    Size: 5
Max index: 4
vChar2: 5 4 3 2 1
END char TESTING

BEGIN int TESTING
vInt: 6    Size: 1
vInt: 3 6    Size: 2
vInt: 3 6 2    Size: 3
vInt: 3 2 -1610612736    Size: 3
vInt: 4 2 -1610612736    Size: 3
vInt1: 4 2 -1610612736
vInt1: 4 2 8
vInt2: 9    Size: 1
vInt2: 1 2 3 4 5
Size: 5
Max index: 4
v2: 5 4 3 2 1
END int TESTING

BEGIN double TESTING
vD: 6.131    Size: 1
vD: 3.33 6.131    Size: 2
vD: 3.33 6.131 2.1    Size: 3
vD: 3.33 2.1 0    Size: 3
vD: 4 2.1 0    Size: 3
vD1: 4 2.1 0
vD1: 4 2.1 8.42
vD2: 9.99    Size: 1
vD2: 1.3 2.4 3.6 4.2 5.13
Size: 5
Max index: 4
vD2: 5.13 4.2 3.6 2.4 1.3
END double TESTING
```