COMP 1602 – Computer Programming II

2023/2024 Semester 2

Lab #12

Sorting and Searching
Review of Coursework Exam #2

1. The array, `num`, needs to be sorted in *ascending* order.

num	32	8	29	6	17	32	3	12	8	23

Starting with the contents of `num` as shown above, *draw* the modified array:

a) After each of the first five (5) passes of selection sort;
b) After each of the first five (5) passes of insertion sort;
c) After each of the first five (5) passes of bubble sort.
d) If the array was already sorted, would insertion sort or bubble sort be more efficient?

Show all working.

2. The array below is sorted in *descending* order.

34	32	29	23	17	11	10	6	5	1

Using the binary search algorithm, how many comparisons are needed to find the following keys?

a) 29
b) 50

Show all working.

3. A sorted integer array contains the following values:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
22	25	42	47	47	50	62	65	71	74	80	82	83	87	88	93	97

(a) A *binary search* is used to search for the value 88 in the array. How many searches will be made before it is found? Show your working.

(b) Suppose that a *linear search* is made.
i) How many searches will be made to find 88?
ii) Does a linear search perform better on a sorted array or an unsorted array? Explain your answer.

(c) Suppose a modified binary search function returns the *location* where a value should be inserted in a sorted integer array, if it is not present. The prototype of the modified *binary search* function is as follows:

```
int modBinarySearch (int A[], int numeElements, int key);
```

Write a segment of code which inserts a value *key* in the sorted integer array *A*. The number of elements in the array is *numElements***.** Assume that there is enough space to accommodate the new value *key*.

Coursework Exam #2

Question 3.

a. A text file, **snack.txt**, has the following format:

Snack ID (int)
Snack Name (C-String)
Snack Type (char)
IngredientID Calories (Ingredient struct)
IngredientID Calories
IngredientID Calories

An example of the text file is as follows:
1001
Potato_Chips
U
10111 201
10112 45
10113 87
-1

Assume there is only data on one snack and the **`Ingredient`** struct is defined as:

```
struct Ingredient {
      int id;
      int kcal;
};
```

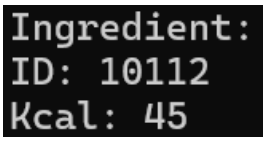
You are required to write C++ code to read the text file, store the data in the specified data types, then write the data to a binary file, **`snack.bin`**.

Please note the following:

- The data are to be stored in the same order as they appear in the text file.
- You do not need to check for errors on opening the binary file.
- Assume the integer -1 signifies the end of data in the text file. You do not need to write this to the binary file.
- Assume the **`ifstream txtFile`** has already been opened.

[8 marks]

b. Write a C++ function, **`seekAndRead()`**, which seeks to the second **`Ingredient`** structure in the **`"snack.bin"`** file, and prints the details of the structure to the screen. You may start seeking from the end of the file. Your function should print the following:



[5 marks]

Q3 Total Marks: 13

Question 4.

a. Given the following C++ struct **`Device`**:

```
struct Device {
    string name;
    string brand;
    double price;
    char tier;
};
```

Assume you have an array of **`Device`** structures called **`inventory`**, which is sorted in ascending order by the **`price`** field.

Write a C++ function named **`findDeviceByPrice`** that implements the binary search algorithm to find a device by its **`price`** in the **`inventory`**. Your function should adhere to the following specifications:

- The function should take three arguments:
  - An array representing the inventory of **`Device`** structures.
  - An integer, **`size`**, representing the size of the array.
  - A double, **`price`** representing the target price to find.
- The function should return an **`int`**:
  - The index of the **`Device`** in the **`inventory`** if a device with the exact target price is found.
  - **`-1`** if there is no device with the target price in the **`inventory`**.

[6 marks]

b. Explain the main differences between a binary search algorithm and a linear search algorithm. Your explanation should include at least two key points of comparison.

[2 marks]

Question 4 continues on next page…

c. Given the following C++ code, clearly indicate what is printed in the **`foo()`** method, and what is printed in the **`main()`** method:

```
int foo (int a) {
    int * b;

    b = &a;
    *b = *b + 45;

    cout << "a = " << a;
    cout << " and *b = " << *b;
    cout << endl;

    return a;
}

int main() {
    int a = 30;
    int b;

    b = foo (a);

    cout << "a = " << a;
    cout << " and b = " << b;
    cout << endl;

    return 0;
}
```

[4 marks]

Q4 Total Marks: 12

End of Lab #12