Search algorithms

A – Theory part

A.1. Discuss the **efficiency** and **inefficiency** of *linear search*. What are the **best case** and **worst case** of *linear search*. Repeat the question for *binary search*.

A.2. Identify the **key difference(s)** between *linear search* and *binary search*.

A.3. *Linear search* must examine, on average, half of the N elements of an array while *binary search* needs to only check $\log_2 N$. Thus, *binary search* is a great deal more efficient than *linear search*. Why, then, should we still study and use *linear search*?

A.4. The efficiency of *linear search* could be improved by adding a sentinel. Explain why.

A.5. Consider the following array of integers, {1, 4, 6, 7, 9, 10, 14}. For *linear search*, which elements will be examined to determine that 9 is in the array, and which elements will be examined to determine that 11 is not in the array? Repeat the question for *binary search*.

A.6. Fill in the blanks in the following code segment to implement *sequential search* on the given array. If the number is found, the function should return its array subscript. Otherwise, −1 is returned indicating the value did not appear in the array.

```
int searchList(const int list[], int numElems, int value) {
    int index = ________; // Used as a subscript to search array
    int position = -1;    // To record position of search value
    bool found = false;   // Flag to indicate if the value was found
    while ( ______________________ && ____________________) {
        if (__________________) {      // If the value is found
            found = true;      // Set the flag
            position = index;       // Record the value's subscript
        }
        index++;            // Go to the next element
    }
    return ___________________;
}
```

A.7. Fill in the blanks in the following pseudo-code segment to implement a search algorithm of the given array. Which search algorithm is it?

```
array people[5]
```

```
people = [“Imogen”, “Fletcher”, “Kirstie”, “Zoe”, “Gavin”]
found = False
x = 0
searchfor = inputName()
while found = False ………… x < 5
    if people[x] = searchfor then
        found = ……………
        print “found at position” + …………
    else
        x = x + 1
    endif
endwhile
```

A.8. Consider the following array of strings, `[“Imogen”, “Fletcher”, “Kirstie”, “Zoe”, “Gavin”]`. Is the given array could be searched using *binary search*? Justify your answer.

A.9. Fill in each of the following blanks with appropriate indexical values.

- *You are using binary search to look for the number 28 in the array* {5, 11, 25, 28, 45, 78, 100, 120, 125}*. When the algorithm stops, the value of the variable first (or left) is* ____________ *and the value of the variable last (or right) is* ___________.
- *You are using binary search to look for the number 50 in the array {*11, 12, 16, 18, 23, 29, 33, 48, 54, 57, 68, 77, 84, 98*}. When the algorithm stops, the value of the variable first (or left) is* ____________ *and the value of the variable last (or right) is* ___________.

A.10. Fill in each of the following blanks with an appropriate number.

- *32 teams qualified for the 2018 World Cup. If the names of the teams were arranged in sorted order (an array), the binary search would have to examine at most* _________ *items to find the location of a particular team in the array.*
- *In 2013, there were 193 member states in the United Nations. If the names of these states were sorted alphabetically in an array, the binary search would have to examine at most* __________ *names to locate a particular name in the array.*
- *Given an array of prime numbers that are from 2 to 311 in sorted order: {2, 3, 5, 7, 11, 13, ..., 307, 311}. There are 64 items in the array. The binary search would have to examine about* _______ *items before concluding that 52 is not in the array, and therefore not prime.*
- *The 2014 "Catalogue of Life" contains about 1,580,000 names of species. If these names were sorted in an array, in the worst case, how long would it take to use binary search to find the name of a particular species in the array?* __________.

B – Coding part

For each of the below three requirements, B.1 – B.3, consider both cases: the array is unsorted, and the array is already sorted.

B.1. Write the function, `remove`, to take three parameters: an array of integers, the number of elements in the array, and an integer (say, `removeItem`). The function should find and delete the first occurrence of `removeItem` in the array. If the value does not exist or the array is empty, output an appropriate message. (Note that after deleting the element, the number of elements in the array is reduced by 1.)

B.2. Write the function, `removeAt`, to take three parameters: an array of integers, the number of elements in the array, and an integer (say, `index`). The function should delete the array element indicated by `index`. If `index` is out of range or the array is empty, output an appropriate message. (Note that after deleting the element the number of elements in the array is reduced by 1.).

B.3. Write the function, `removeAll`, to take three parameters: an array of integers, the number of elements in the array, and an integer (say, `removeItem`). The function should find and delete all the occurrences of `removeItem` in the array. If the value does not exist or the array is empty, output an appropriate message. (Note that after deleting the element, the number of elements in the array is reduced.)

B.4. Your state is in a process of creating a weekly lottery. Once a week, five distinct random integers between 1 to 40 (inclusive) are drawn. If a player guesses all the numbers correctly, the player wins a certain amount. Write a program that does the following:

a) Generate five distinct random lottery numbers between 1 to 40 (inclusive) and stores them in an array.

b) Sort the array containing the lottery numbers.

c) Prompt the player to select five distinct integers between 1 and 40 (inclusive) and store the number in an array. The player can select the numbers in any order and the array containing the numbers need not to be sorted.

d) Determine whether the player guessed the lottery numbers correctly. If the player guessed the lottery number correctly, output the message “You win!”; otherwise, output the message “You lose!”, as well as the lottery numbers.

Your program should allow a player to play the game as many times as he wants to play. Before each play, generate a new set of lottery numbers.