

CST 370 – Spring 2020
Homework 2
Due: 02/07/2020 (Thursday) (11:55 PM)

Name: _____ Bobby Davis _____

ID: _____ 1806 _____

How to turn in?

- Write your answer to the question 1 and 2 and submit it on the iLearn. Note that we **accept only a PDF** file. Do not submit a different file format. Also, don't forget to write your name and class ID at the top of your homework document.
- For the questions 3 and 4, you should submit two source programs (**hw2_1.cpp** and **hw2_2.cpp**) on the iLearn. Note that you can use the Java for the homework such as **hw2_1.java** and **hw2_2.java**.
- Thus, you have to submit three files (one PDF file and two source files) on the iLearn.

1. Assume that you should search a number in a list of n numbers. How can you take advantage of the fact that **the list is known to be sorted**? Give separate answers for the following two cases.

(a) A list represented in an array.

In a sorted array you can use a binary search to find a number very quickly.

(b) A list represented in a linked list.

In a linked list the fastest search time is always linear so there is no real advantage. Although if we had a doubly linked list and we new we were searching for a really big number then it might make sense to start from the end to try and find the number faster.

2. Assume that you have **10** identical-looking balls and a two-pan balance scale with no weights. One of the balls is a fake, but you don't know whether it is lighter or heavier than the genuine balls, which all weigh the same. Describe your idea to determine in the minimum number of weighings whether the fake ball is lighter or heavier than the others. In the problem, you **don't need to identify the fake ball itself**. It's good enough for you to **just say "It's heavier or lighter"**. Present the minimum number of weighings and your answer clearly.

The answer is 2 weight measurements.

First, I would separate the 10 balls into 3 separate groups of 3, 3, and 4. Then I would weigh the two groups of 3. There would be 2 cases to consider:

Case 1: The two groups of 3 are even.

This would prove that all six of the balls I just weighed are all the same weight, so the fake ball must be in the group of 4 I set aside. Next, I would pull 4 balls out of the 6 even balls and weigh them against the 4 I set aside. If the 4 that were initially set aside goes down, then the ball is heavier and if they go up the ball is lighter.

Case 2: The two groups of 3 balls are uneven where one side rises and the other falls.

If this happens then I know the fake ball must be in one of these two groups of 3 and that the 4 I set aside must all be the same weight. Then I can take 3 from the group of 4 I set aside and weigh them against either one of the groups of 3. If I weigh the 3 balls, I know

are even against the group of 3 balls that went down and they are even, then the fake ball must be lighter. If they are not even, then the fake ball must have been heavier.

=====

This is the HackerRank link: <https://www.hackerrank.com/cst370-s20-hw2>

3. Write a program called **hw2_1.cpp (or hw2_1.java)** that reads a positive integer number from a user and displays the reverse of the number. For the program, you can assume that the input number is in the range of the typical “int” data type.

Sample Run 0: Assume that the user typed the following number.

1234321

This is the correct output.

1234321

Sample Run 1: Assume that the user typed the following number.

425

This is the correct output.

524

Sample Run 2: Assume that the user typed the following number.

1200

This is the correct output. Note that the reverse of 1200 is not 0021. It should be 21 because we should remove the leading zeros.

21

4. Write a program called **hw2_2.cpp (or hw2_2.java)** that reads two timestamps of two events from a user and displays the difference between the two timestamps. For the program, you can assume that each timestamp is composed of the hour (0 ~ 23), minute (0 ~ 59), and second (0 ~ 59) format. Your program should present the difference from the second event (= second timestamp) to the first event (= first timestamp). Note that **the second event always happens after the first event** and your program should display the time difference of the events.

Sample Run 0: Assume that the user typed the following two lines.

```
18:45:30
20:50:59
```

This is the correct output of your program.

```
02:05:29
```

Sample Run 1: Assume that the user typed the following two lines.

```
20:18:59
04:25:17
```

This is the correct output of your program.

```
08:06:18
```

Sample Run 2: Assume that the user typed the following two lines.

```
02:00:25
15:30:00
```

This is the correct output of your program.

```
13:29:35
```