Student Name CHETAN ARORA Student Number 100976240 UserID carora18 @mySeneca.ca

Special Notes to Instructor Click or tap here to enter text.

There are many notes in the instructions to help you earn marks for the questions below.

Exercise One of Two – **integer overflow** (80 points)

1) ➔ (7.5 points)) If a variable counting seconds is stored in a signed **long** 32-bit integer, how many **days** will it take until that integer overflows? (to one decimal place)

A 32-bit integer can store value from -2,147,483,648 to 2147483647. If a variable counting seconds is start from 0 and incremented by 1 (every second), it will take 2147483648 seconds which is equavalent to 24855 days approximately.

2) ➔ (15 points) Convert the maximum value of an unsigned **long** 32-bit integer, representing hundredths of a second, into whole numbers of

days : hours : minutes : seconds . hundredths of a second.

After *n* days, how many hours remain? After *n* hours, how many minutes remain? etc.

497	**2**	**27**	**52**	**95**

DAYS HOURS : MINUTES : SECONDS .HUNDREDTHS

4294967295	INT _MAX for 32-bit unsigned long integer		
497	Days	8640000	Hundredths of seconds in a day
887295	remaining hundredths		
2	Hours	360000	Hundredths of seconds in a hour
167295	remaining hundredths		
27	Minutes	6000	Hundredths of seconds in a minute
5295	remaining hundredths		
52	Seconds	100	Hundredths of seconds in a minute
95	remaining hundredths		
95	Hundredths	1	Hundredths of second

3) ➔ (2.5 points) What are the maximum and minimum values that can be stored in a **short** 16-bit signed integer?

```
16-bit signed integer maximum = 32767 … minimum =-32768
```

4) ➔ (5+5 points) Give examples of two **short** 16-bit signed integers that when added together would cause overflow.

```
 25000 +  25000 are two positive values causing overflow when added together.
```

```
-25000 + -25000 are two negative values causing overflow when added together.
```

Binary Search Bug

5) ➔ (10 points) What is potentially wrong with the **(low + high) / 2** calculation to find the middle point? Under what conditions would the calculation go wrong?

The calculation (low+ high)/ 2 to find middle point is potentially wrong when the sum of values of low and high cannot hold by the data type which is used to perform the above calculations. This generally happen when low and high are very large positive or negative numbers and their sum exceed the maximum limit of data type.

6) ➔ (10 points) REWRITE the `mid` calculation to prevent overflow

from `mid = (low + high) / 2;`

to **`mid = (low /2 )+ (high/2);`**

7) ➔ (25 points) Write a 250+ word "reflection"(similar to a workshop in your programming class) describing the steps you used to develop and test your solution to the calculation bug.

I followed the following steps to develop and test your solution to the calculation bug:

1) Identify the problem: The first step was to find the problem with the formula (low + high) / 2 . I search the issue and found that the calculation (low+ high)/ 2 to find middle point is potentially wrong when the sum of values of low and high cannot hold by the data type which is used to perform the above calculations. This generally happen when low and high are very large positive or negative numbers and their sum exceed the maximum limit of data type.

2) Find a solution: In next step, I would be to find a solution to the problem. I found that this issue can be solved by using the formula low/2 + high/2 instead of (low + high) / 2 . This ensures that the calculations are always rounded down to the bottom, even when low + high becomes negative.

3) Implemented solution: Once I had the solution, I implemented it in my code. To ensure that the division is always rounded to the bottom boundary, I replaced the (low + high) / 2 formula with low/2 + high/2.

4) Testing the solution: The final step was to test the solution to ensure it worked as expected. I tested the code using visual studio (Tool by Microsoft) with different minimum and maximum

values to ensure that the mean was always calculated correctly. I also tested the code with more or less values to see if the overflow issue was fixed.

Exercise Two of Two – **Numbering Systems and Conversions (20 points)**

8) ➔ (10 points) What is the hex value for these colours?

Red decimal	Green decimal	Blue decimal	Hex triplet	Colour Description
15	245	231	#0FF5E7	Bright Aqua
192	255	238	#C0FFEE	Humming Bird
208	13	30	#D00D1E	Strong red
186	187	30	#BABB1E	Rio Grande
126	164	112	#7EA470	Harmonies

9) ➔ (10 points) Fill in this chart as per the column headings

Hex triplet	Red decimal	Green decimal	Blue decimal	Describe the Final Colour *and* change the cell's background colour, i.e. R-click and see MS Word 'Shading'
#302432	48	36	50	⬛
#204C02	32	76	2	🟩
#D64A53	214	74	83	🟥
#404891	64	72	145	🟦