Abcedi Ilacas138180211 ailacas1@mySeneca.ca  
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. (5 points) Using your student number as the number of **seconds**,  
 how many **days** will it take until that value is reached? (to two decimal places)

1599.31

🡺 2. (20 points) Using your student number as the number of **seconds**,  
 how much **time** would that represent?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **4** | **139** | **7** | **23** | **31** |
| **YEARS** | **DAYS** | **HOURS** | **MINUTES** | **SECONDS** |

🡺 3. (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 points) Give examples of two **short** 16-bit signed integers that when added together would cause overflow.

 16384 +  16384 are two positive values causing overflow when added together.

-16385 + -16385 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 expression (low + high) / 2 to calculate the middle point in a binary search can potentially lead to integer overflow if the sum of low and high exceeds the maximum representable value for the given data type. The calculation would go wrong when low + high exceeds the maximum representable value for the data type of low and high.

6) 🡺 (10 points) REWRITE themidcalculation to prevent overflow*from*mid = (low + high) / 2;*to*  **mid = low + (high – low) / 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.

In programming, straightforward approaches can sometimes be flawed, as highlighted by the midpoint calculation in binary search using (low + high) / 2. The issue emerged when testing with values close to integer limits, revealing potential overflow when summing low and high.

Step 1: Problem Identification

Tests on larger datasets showed inconsistencies in outputs. This led to pinpointing the overflow issue when low and high summed beyond data type limits.

Step 2: Understanding the Issue

A dive into integer representation clarified the limitations of computer arithmetic. Even if mathematically sound, formulas can be constrained by practical system limits.

Step 3: Solution Brainstorming

With the problem identified, a new formula was sought to prevent overflow. The modified formula, low + (high - low) / 2, emerged as a robust solution.

Step 4: Implementation & Testing

After implementing the new formula, rigorous testing confirmed its correctness across various datasets, ensuring no overflow instances.

Step 5: Reflection & Documentation

This exercise stressed the importance of thorough testing and adapting to system constraints. Documenting this change is essential for future reference and understanding.

In summary, this wasn't merely a bug fix, but a lesson on the subtleties of software development, the significance of testing, and the need for adaptability.

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

8) 🡺 (5 points ) Fill in this chart as per the column headings

|  |  |  |  |
| --- | --- | --- | --- |
| Red decimal | Green decimal | Blue decimal | **Hex triplet** |
| 192 | 255 | 238 | #C0FFEE |
| 126 | 164 | 112 | #7EA470 |
| 186 | 187 | 30 | #BABE1E |
| 208 | 13 | 30 | #D00D1E |
| 15 | 245 | 231 | #0FF5E7 |

9) 🡺 (15 points)Fill in this chart as per the column headings

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Hex triplet | **Red decimal** | **Green decimal** | **Blue decimal** | Change each row's Font or Shading colour to match the RGB value |
| #3A26F2 | 58 | 38 | 242 | 🡨 Font or Shading colour matches RGB |
| #844D42 | 132 | 77 | 66 | 🡨 Font or Shading colour matches RGB |
| #094353 | 9 | 67 | 83 | 🡨 Font or Shading colour matches RGB |
| #7D6CB7 | 125 | 108 | 183 | 🡨 Font or Shading colour matches RGB |