*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)

It will take 24855.1 days until that integer overflows.

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** | **11952** | **717748** | **42,949,672** | **4,294,967,295** |
| **DAYS** | **HOURS :** | **MINUTES :** | **SECONDS .** | **HUNDREDTHS** |

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 = 32,767 … minimum = -32,768

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

 16879 +  17623 are two positive values causing overflow when added together.

-16263 + -18479 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?

By using (low +high) / 2 potentiallly the sum of both the integers that is the value of (low + high) can be greater than the maximum value that can be stored in the memory for that data type and lead to program overflow.

The calculation would go wrong if the value of (low + high) is greater than the memory allocation for that data type.

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.

When we are finding the middle point between any two integers, there is a potential mistake that can occur while using the formula: (low + high) / 2. This happens because even though two numbers can be individually inside the memory constraints of a data type, but when the are added if the sum is more than the memory allocated for that data type, it can lead to overflow. So, inorder to change it and fix the bug, we have to change the formula used.

Instead of using mid = (low + high) / 2, I have used mid = low + (high – low)/2 where using the rule of BODMAS, operations inside the brackets take the first priority followed by division operator and then followed by addition. By using this formula the arithematic values and opearations remain the same, but the way/order of calculation changes which leads to fixing of the bug. The reason why this doesn’t lead to overflow is because while subtracting the low value from the high value the answer or result will always be less than that of the high value which is already inside the memory constraint of that data type.

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 Turquoise |
| 192 | 255 | 238 | #C0FFEE | Arabica Mint |
| 208 | 13 | 30 | #D00D1E | Strong Red |
| 186 | 187 | 30 | #BABB1E | Rio Grande |
| 126 | 164 | 112 | #7EA470 | Frog Hollow |

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 | Dark Shade of Magenta |
| #204C02 | 32 | 76 | 2 | Dark Lawn Green |
| #D64A53 | 214 | 74 | 83 | Fire Engine Red |
| #404891 | 64 | 72 | 145 | Dark Moderate Blue |