Yuvraj Singh155580210 yuvraj-singh5@mySeneca.ca

**Exercise 1** of 3 – **integer overflow** (55 points)

1a) If a variable counting hundredths of a second is stored in a signed **long** 32-bit integer,   
how many **days**, to two decimals, will it take until that integer overflows? **(5 points)**

It will take 248 days for a 32-bit integer to overflow with given conditions.

1b) Convert the maximum value in a signed **long** 32-bit integer,   
 representing hundredths of a second,   
 into days, hours, minutes, seconds, and hundredths of a second? **(7.5 points)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **248** | **5965** | **357913** | **21474836** | **2147483647** |
| **DAYS** | **HOURS** | **MINUTES** | **SECONDS** | **hundredths** |

1c) What are the maximum and minimum values that can be stored in a **short** 16-bit signed integer? **(2.5 points)**

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

1d) Give examples of two **short** 16-bit signed integers that when added together would cause overflow. **(5 points)**

 16,384 +  16,385 are two positive short integers causing overflow when added together.

-17,563 + -17,003 are two negative short integers causing overflow when added together.

Binary Search Bug

1d) What is potentially wrong with the **(low + high) / 2** calculation to find the middle point? Under what conditions would the calculation go wrong? **(****5 points)**

If we use (low + high) / 2, there is always a risk of an overflow because if the number which is being added gets bigger than the limit of a computer (let’s assume a 64-but operating system), it will result in an overflow if it passes the highest number limit of the OS. And thus, we get different results than expected.

1e) REWRITE the code to prevent overflow **(10 points)***from* **mid = (low + high) / 2;***to* mid = low + (high – low) / 2; ;

1f) **(20 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 binary search bug.

To get a result without any errors, I tried quite a few steps to get an output with zero error. Firstly, I opened the midBugTest.c in my visual studio. By default, the mid formula was mid = ( low + high ) / 2; . But as this formula is not feasible to use in scenarios like this, because it overflows easily, I though of a different formula which has much more flexibility and will not overflow as easily. The formula is mid = low = + (high – low) / 2; . Actually, the case here is that the numbers exceed the limit of a computer system if we add it directly and then divide it to 2. Instead we need to find something which can work while being under the limit of the highest number of the computer system. I noticed with mid = (low + high) / 2; that it gave me the resluts for half of the total numbers i.e – 53/106 so that means that the formula overflows as the addition of 2 numbers is out of the maximum limit of the computer. So to overcome that problem, I used mid = low + (high – low) / 2; because in this particular formula, the two numbers do not add up directly and thus, do not go out of the highest limit avoiding an overflow. During my high school, I learnt the concept of binary search and using it to find the middle number when I was learning python language, so I remembered the formula since then and I applied it here. After replacing the new formula, there were 0 errors and it was working as expected.

**Exercise 2** of 3 – **Boolean logic** (**35 points**)

**2. For any given date, what is the** **Boolean logic to decide if you should attend school during the current term?**

Your C language **if** statement(s) go here. There is no need to write a program.

If ((today >= startClasses && today <= endClasses && today <= startStudyWeek && today >= endStudyWeek) ( today != familyDay,goodFriday,victoriaDay,canadaDay,civicDay,thanksgivingDay))

printf(“You have school today”);

**Exercise 3a and 3b** of 3 – **Numbering Systems and Conversions (10 points)**

3a) What is the hex value for these colours? **(5 points)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Red decimal** | **Green decimal** | **Blue decimal** | **Hex triplet** | **Colour Description** |
| 208 | 13 | 30 | #D00D1E | Red |
| 126 | 164 | 112 | #7EA470 | Asparagus |
| 186 | 187 | 30 | #BABB1E | Key lime pie |
| 15 | 245 | 231 | #0FF5E7 | Bright turquoise |
| 192 | 255 | 238 | #C0FFEE | Aero blue |

3b) Fill in this chart as per the column headings **(5 points)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **6 digit Hex code** | **Red decimal value (0-255)** | **Green decimal value (0-255)** | **Blue decimal value (0-255)** | **Describe the Final Colour *and* change the cell's background colour, i.e. R-click and see MS Word 'Shading', to match the values for RGB** |
| #D64A52 | 214 | 74 | 82 | LIGHT MAROON |
| #404892 | 64 | 72 | 146 | NAVY |
| #302431 | 48 | 36 | 49 | BLACK |
| #204C04 | 32 | 76 | 4 | DARK GREEN |