🡺 Your name: Amirhossein Sabagh Student No.: 152956199

UserID: [asabagh@mySeneca.ca](mailto:asabagh@mySeneca.ca)

**Activity 1** of 3 – **integer overflow**

🡺 If the timer value is stored in a signed **long** 32-bit integer,   
how many **days**, to two decimals, will it take until that integer overflows? **(5 points)**

2,147,483,647 / 8,640,000 = 248.55 -> it will take 248 days for the counter to overflow

🡺 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 = 32,767 … minimum = -32,768

🡺 Give examples of values that would cause overflow in positive and negative directions when two **short** 16-bit signed integers are added together. **(5 points)**

17,000 + 16,000 are two positive **short** values causing overflow when added together.

-15,569 + -18,321 are two negative **short** values causing overflow when added together.

🡺 What is potentially wrong with the **(low + high) / 2** calculation to find the middle point? Under what conditions would the calculation go wrong? **(7.5 points)**For a demonstration of the problem, run **MidBugTest.exe** found in this week's zip file.

The intermediate value of (low + high) can overflow. How?

Let's say we have the low value of 1,000,000,000 and the high value of 2,000,000,000. The sum of low and high (3,000,000,000) would overflow on a long 32-bit integer. The result of calculation would be (- 852,516,353‬ / 2) = - 426,258,176.5

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

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

it can also be done using " mid = low + ((high - low) / 2);" Not to lie, I searched the net and found this :)

🡺 Describe the steps you used to develop and test your solution to the binary search bug. **For the full 20 points**, what were the details of your process from problem analysis to solution implementation? (This is like the reflection component in your C course workshop.)

So apparently my solution, dividng each of the low and high values to 2 and then add them together, worked but there was still some errors (number of errors reduced from 53 to 22).

The other method worked perfectly and there were no errors.

I didn't really get how this method would avoid the overflow at first. So I went and searched for the reason and found this comment on Quora ( I could have paraphrased it and used it as mine but what benefit would that have really. It was better to just share it because sharing is caring).

<https://qr.ae/pNKsTJ>

**Activity 2** of 3 – **Boolean logic (25 points)**

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

[I didn't completely get the question, so I just do my best] // this is a mix of pseudocode and c programming code :)

int today;

long startClasses = 20200519, endClasses = 20200814, CanadaDay = 20200701, CivicHoliday = 20200802;

printf("Enter today's date in the right formaet (YYYYMMDD) : ");

scanf("%d", today);

if ( today >= startClasses && today <= endClasses && today != CanadaDay && today != CivicHoliday && today !=Saturday && today != Sunday && today != Monday) // I couldn't use classesDoW[ ], so I will assume that there are some classes on every other day

{

printf("Go to school, you lazy boy");

}

else printf("Stay home and Chill");

**Activity 3a and 3b** of 3 – **Numbering Systems and Conversions (15 + 10 points)**

3a 🡺 What is the hex value for these colours? **(15 points)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Colour** | **Red value** | **Green value** | **Blue value** | **6 digit Hex code** |
| White | 255 | 255 | 255 | # FFFFFF |
| **Grey** | 128 | 128 | 128 | # 808080 |
| **Black** | 0 | 0 | 0 | # 000000 |
|  | 0 | 211 | 023 | # 00D317 |
|  | 103 | 110 | 160 | # 676EA0 |

3b 🡺 Fill in this chart as per the column headings **(10 points)**

[P.S: I suck at colors. Describing colors was harder than the other parts of this activity for me :) ]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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** |
| #FB6905 | 251 | 105 | 005 | ORANGE |
| #0AA2C8 | 10 | 162 | 200 | OCEAN BLUE |
| #E160B7 | 225 | 96 | 183 | PINK (or Purpule) |
| #0B6A0C | 11 | 106 | 12 | DARK GREEN |