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

1. 🡺 What is potentially wrong with that line of code? **(10 points)**

The issue with this formula is that, the addition causes the intermediate result to possibly surpass the highest possible value that can be stored in that data type. If the intermediate result does surpass the data type’s highest value, the variable overflows and wraps to the lowest possible value that data type can store. This could be zero if unsigned or else if it is signed, it could be result in a negative number which is often disastrous when searching arrays, since searching a negative index value in an array causes a sematic error since in C, element’s indexes start at the number zero. Searching a negative index in an array could cause you to search a different part of memory which could be disastrous.

1. 🡺 What values for **(low + high)** would result in an overflow? **(10 points)**[play with the integerOverflow.exe program in the zip file]

For a 2-byte signed variable (short), any value greater than 16 384, to be stored in the high variable, could potentially cause an overflow.

If high is 16 384 and low is 16 382 the intermediate result is 32 766 <- this is good

If high is 16 385 and low is 16 383 the intermediate result is 32 768 <- this will overflow

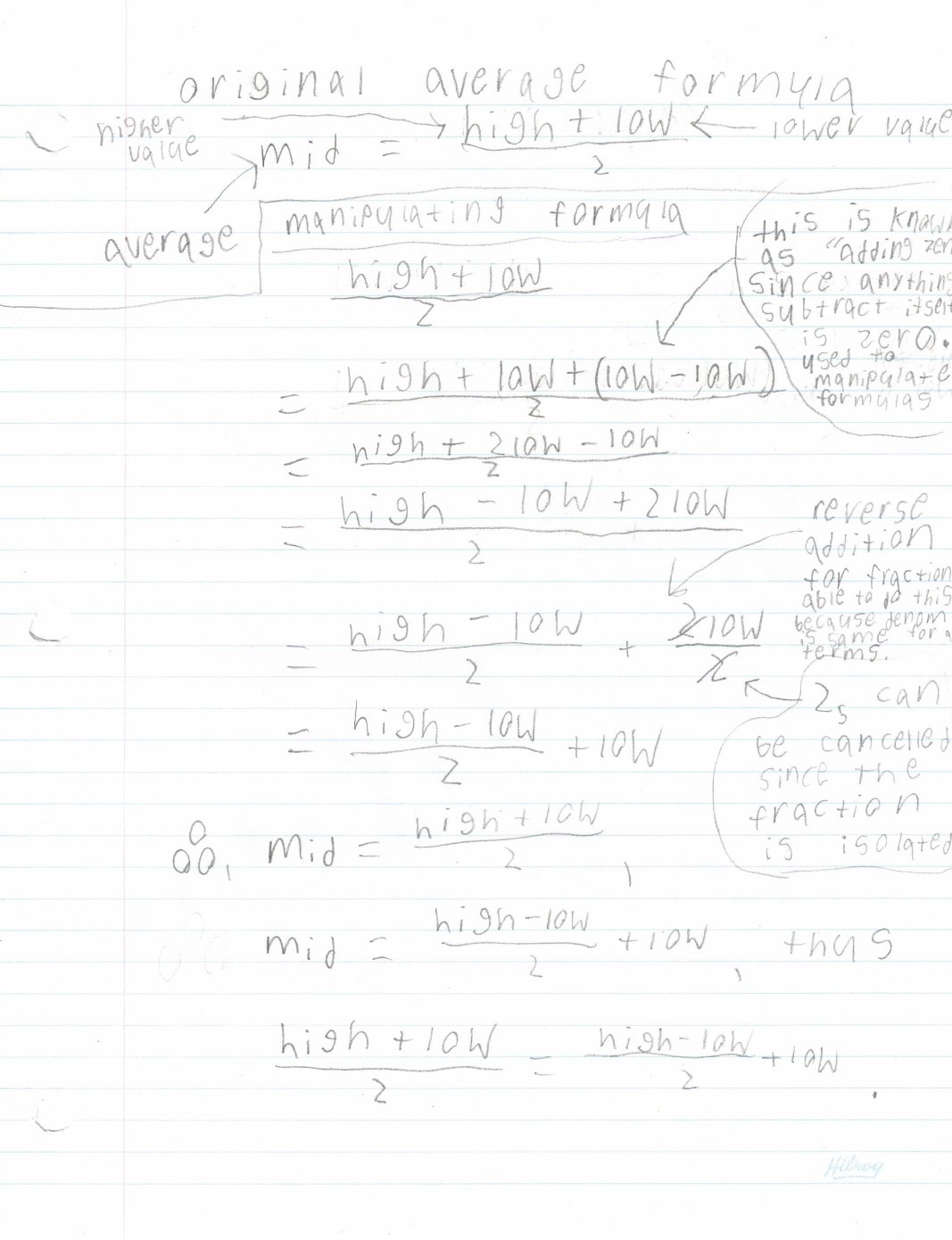
1. 🡺 How would you fix that line of code? **(30 points)**

By manipulating the math formula for the average of two numbers, I was able to create the formula:

Mid = low + ((high - low) / 2)

Proof is on the next page.

This formula works because it subtracts, you never have to worry about the intermediate value overflowing since the intermediate value will never be greater than the high value. Furthermore, since the formula subtracts a lower number from a larger number it will never be a negative result, so we also do not have to worry about the overflowing occurring in the other direction.



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

On any given date, how do you decide whether you have to go to school for a class?

**🡺 What is the Boolean logic to determine IF you should go to school for a class during the current term on any given date?**

**Int main(void) {**

**If ((DoW == Monday || DoW == Tuesday || DoW == Wednesday || DoW == Thursday || DoW == Friday) && (!isHoliday(today) || !isBreak(today))) {**

**goToClass();**

**}**

**}**

**// isHoliday() is a function that determines if the date passed as an argument is equal // // to any holiday date will return a non-zero number if true.**

**//Since today stores the current date it can be a struct which carries the day, month, // and year which all can be accessed using the dot operator.**

**Int isHoliday(today) {**

**if (today.month == October && (today.dayNum – 7 <= 7)) return 1; // thanksgiving determination algorithm**

**else if (today.month == December && today.dayNum == 25 || today.dayNum == 24) return 1**

**// Christmas and Christmas Eve**

**else if (today.month == December && today.dayNum == 31) return 1 // New Year’s Eve**

**else if (today.month == January && today.dayNum == 1) return 1 // New Year’s Day**

**…**

**else return 0;**

**// nothing matches**

**}**

**// isBreak() is a function that determines if the date passed as an argument is equal to //any break date during the term such as study week. Will return a non-zero number if //true.**

**int isBreak(today) {**

**if(today.month == October && (today.dayNum >= 22 && today.dayNum <= 26))**

**return 1;**

**//Study Week**

**…**

**else return 0;**

**// nothing matches**

**}**

**Activity 3** of 3 – **Numbering Systems and Conversions (25 points)**

🡺 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 |
| Teal | 0 | 128 | 128 | #008080 |
| Grey | 128 | 128 | 128 | #808080 |
| Purple | 128 | 0 | 128 | #800080 |
| Black | 0 | 0 | 0 | #000000 |

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3 Hex values** | **Red decimal value (0-255)** | **Green decimal value (0-255)** | **Blue decimal value (0-255)** | **Describe the Final Colour** |
| #FF9900 | 255 | 153 | 0 | Orange |
| #789ABC | 120 | 154 | 188 | Grey |
| #FEDCBA | 254 | 220 | 186 | Very light pale colour |
| #00B309 | 0 | 179 | 9 | Dark green |