COIS 2300H Assignment 2

This assignment is worth 10% of your final Grade,

# Arithmetic Questions (A-G 0.5 marks each)

**Do these questions by hand, show your steps. Check your answer with a calculator so you don’t get it wrong. If there is a Sri/Alaadin Question, you will only get marks for that part.**

## **A:**

**3.1 <§3.2> What is 5ED4 - 07A4 when these values represent unsigned 16- bit hexadecimal numbers? The result should be written in hexadecimal. Show your work. Sri/Alaadin addition: what is A81F -5D2E**

**5ED4 – 07A4   
 0101 1110 1101 0100  
-0000 0111 1010 0100  
--------------------------------  
0101 0111 0011 0000  
 5 7 3 0  
-------------------------------  
  
5ED4 – 07A4 = 5730**

**A81F – 5D2E**

**1010 1000 0001 1111  
-0101 1101 0010 1110  
--------------------------------**

**0100 1010 1111 0001  
 4 10=A 15=F 1  
----------------------------------**

**A81F – 5D2E = 4AF1**

## **B:**

**3.2 <§3.2> What is 5ED4 - 07A4 when these values represent signed 16- bit hexadecimal numbers stored in sign-magnitude format? The result should be written in hexadecimal. Show your work. Sri/Alaadin addition: what is A81F-5D2E**

**Answer :**

**5ED4 – 07A4   
 0101 1110 1101 0100  
- 0000 0111 1010 0100  
--------------------------------  
 0101 0111 0011 0000  
 5 7 3 0  
-------------------------------**

**Answer is 0 101011100110000**

**The first digit of the binary number is 0, which means positive, thus the final answer of signed subtraction in hex is 5730.**

**A81F – 5D2E**

**1010 1000 0001 1111  
-0101 1101 0010 1110  
--------------------------------**

**0100 1010 1111 0001  
 4 10=A 15=F 1  
----------------------------------**

**Answer is 0 100101011110001**

**The first digit of the binary number is 0, which means positive, thus the final answer of signed subtraction in hex is 4AF1.**

## **C:**

**3.20 <§3.5> What decimal number does the bit pattern 0×0C000000 represent if it is a two’s complement integer? An unsigned integer? ( that is 0x0C000000 if you need a different font to read it). Sri/Alaadin: Same question but for 0x268b300c**

**Answer :**

## **D:**

**3.21 <§3.5> If the bit pattern 0×0C000000 is placed into the Instruction Register, what MIPS instruction will be executed? ( that is 0x0C000000 if you need a different font to read it) Sri/Alaadin: Same question but for 0x268b300c**

## **E:**

**3.22 [<§3.5> What decimal number does the bit pattern 0×0C000000 represent if it is a floating-point number? Use the IEEE 754 standard. ( that is 0x0C000000 if you need a different font to read it)  
  
F:**

**3.23 <§3.5> Write down the binary representation of the decimal number ~~63.25~~ 31.625 assuming the IEEE 754 single precision format. (Show your steps) – Yes there are tools on the web that will do this for you including steps. You will be asked questions like this on the exams (using a minifloat format) make sure you know how to do it yourself.**

## **G:**

**3.24<§3.5> Write down the binary representation of the decimal number ~~63.25~~  31.625 assuming the IEEE 754 double precision format. (Show your steps)**

# **H: Sri developed theory question 1 mark.**

**(Note, this question is only worth ~ 1 mark, do not over think it). Describe and algorithm for converting a string to a floating-point value. Data from a console is initially read in from a string and must be converted to a floating-point format. Be sure to consider scientific notation as well. A string value of -705.7481 x 1020 for example needs to be parsed in some way to interpret it as an a floating point value .   
A related (but easier problem) is parsing an integer from a string, and you might want to start there (in c++ this is called *atoi* ).**

# **Programming questions**

## **I:**

**You are going to write a MIPS program that does the following (there is C++ sample code below to explain the logic):**

1. **Create an array of 23 INTEGERS, you can put whatever you want in it, but they shouldn’t be sorted. Create procedures to find the largest, and find the smallest. Create another procedure called range, which is the difference between largest and smallest (range should call findLargest and findSmallest)**
2. **Same problem as a) but for floating points (this is a very minor change), but you can make the array size 11.**
3. **Convert your find largest procedure to be recursive (**[**https://www.geeksforgeeks.org/recursive-programs-to-find-minimum-and-maximum-elements-of-array/**](https://www.geeksforgeeks.org/recursive-programs-to-find-minimum-and-maximum-elements-of-array/) **has an example of this in several higher level languages).**

**So something like this C++code, except you need to write it in MIPS**

|  |
| --- |
| **#include "pch.h" //There’s no equivalent to this in MIPS so ignore it**  **#include <iostream> //there’s no equivalent to this either.**  **int theArray[] = { 5, 7, 1, 4, 9, 11, 15, 0, -1 };**  **int findLargest()**  **{**  **int temp = theArray[0];**  **for (int i = 0; i < 10; i++)**  **if (temp < theArray[i])**  **temp = theArray[i];**  **return temp;**  **}**  **int findSmallest()**  **{**  **int temp = theArray[0];**  **for (int i = 0; i < 10; i++)**  **if (temp > theArray[i])**  **temp = theArray[i];**  **return temp;**  **}**  **int findRange()**  **{**  **return findLargest() - findSmallest();**  **}**  **int main()**  **{ std::cout << findLargest() << std::endl;**  **std::cout << findSmallest() << std::endl;**  **std::cout << findRange() << std::endl;**  **}** |

## **J:**

**In a sensible language ( any of C/C++/C#/Java/Python or similar) write a program which correctly calculates add, subtract, multiply and divide using our ‘minifloat’ binary format using an algorithm you code yourself. Some details:**

**Your program only needs to work on two ‘numbers’ at a time, read those in from a text file – failure to read those values from a text file will result in a grade of 0. For each ‘number’ store the sign, exponent and mantissa separately. You can hard code your examples.**

**E.g.**

**Struct minifloat{**

**Int sign; // sign, actually 1 bit**

**Int exponent; // this is the exponent and is 4 bits long (0-15)**

**Int mantissa; //this is the fraction part and is 3bits long (0-7), but I would keep the 1. Stored explicitly, so it becomes 0-15 and 4 digits long.**

**}**

**Show testing for all 4 (add, sub, mul, div) with at least 3 examples (and compare to results you do by hand).**

**Make sure you are error checking for NAN and +/- 0.**

**Note: This is an obviously contrived example, the intent is to force you to think about the algorithm, not to make software that anyone would ever want to use.**