

COIS 2300H Assignment 2:

NAME punyajamishra (please use the name you have on blackboard)

This assignment is worth 10% of your final Grade.

Assignment submission: Print off a copy of this document and your program testing and submit it to Sri's box (just outside OC 102) or under my door/the COIS office door. (If necessary, copy/paste code somewhere so you don't need to print pages of large black boxes from testing).

You must also upload to blackboard a copy of this file, and your source code, zipped (it must be .zip) named for your Trent username.zip to blackboard. E.g. for me that's bsrivastava.zip.

Arithmetic Questions (A-G 2 marks each)

Do these questions by hand, **show your steps**. Feel free to check your answer with a calculator so you don't get it wrong. If you need extra space, expand the boxes.

A: Unsigned Hexadecimal Subtraction (Based on section 3.2 in the text)

Calculate the following values by hand. All numbers represent **unsigned 16-bit hexadecimal numbers**.

Make sure you show your work, including all borrows or intermediate steps, when doing the subtraction. **The result should be written in hexadecimal. Place your answer in the box.**

a)

```
5 E D 4
- 0 7 A 4
-----
0101 1110 1101 0100
-0000 0111 1010 0100
-----
0101 0111 0011 0000
5   7   3   0
-----
```

b)

```
A 8 1 F
- 5 D 2 E
-----
1010 1000 0001 1111
-0101 1101 0010 1110
-----
0100 1010 1111 0001
4   10=A 15=F 1
-----
```

B: Signed Hexadecimal Subtraction (Based on section 3.2 in the text)

Calculate the following values by hand. All numbers represent **signed 16-bit hexadecimal numbers** stored in **sign-magnitude format**. **Make sure you show your work, including all borrows or intermediate steps**. The result should be written in hexadecimal. Place your answer in the box.

a)

```

5 E D 4
- 0 7 A 4
-----
0101 1110 1101 0100
-0000 0111 1010 0100
-----
0101 0111 0011 0000
  5   7   3   0
-----
----- it is positive
since 0 is the leftmost bit. Thus, 5730

```

b)

```

A 8 1 F
- 5 D 2 E
-----
1010 1000 0001 1111
-0101 1101 0010 1110
-----
0100 1010 1111 0001
  4   10=A 15=F 1
-----
----- this is
positive since 0 is left bit. Thus, 4AF1

```

C: Convert to decimal (Based on section 3.5 in the text)

0x0C000000 and 0x268b300c are two's complement integers encoded as a hexadecimal numbers. **Show all intermediate steps** in your work below.

a) In decimal, the value of 0x0C000000 (two's complement) is:

Converting given hex to binary

0000 1100 0000 0000 0000 0000 0000 0000

Doing it's negation

1111 0011 1111 1111 1111 1111 1111 1111

Adding 1 to it

0 1111 0100 0000 0000 0000 0000 0000 → this is the two's complement of the given number

Converting it to decimal the two's complement and it is positive because the original number was positive as it had a 0 in the front

Our final answer is : 201326592

b) In decimal, the value of 0x268b300c (two's complement) is:

converting it into binary

0010 0110 1000 1011 0011 0000 0000 1100

Taking it's negation

1101 1001 0111 0100 1100 1111 1111 0011

Adding 1 to it

0 1101 1011 0111 0100 1100 1111 0100

This is the two's complement and converting to decimal and it is positive because the original number was positive as it had a 0 in the front

Out final answer is : 613101580

D: MIPS Instructions (Based on section 3.5 in the text)

The following questions require converting a hexadecimal number to a MIPS instruction. Be sure to include any source register, destination register, addresses, or immediate values in your answer.

- a) If the bit pattern 0x0C000000 is placed into the Instruction Register, what MIPS instruction will be executed?

On converting to binary we see that the op code is 000011 which is 3 in decimal and that stands for jal and the target is how much to jump which is 0 here. This : jal 0x00000000

—
2

- b) If the bit pattern 0x268b300c is placed into the Instruction Register, what MIPS instruction will be executed?

Binary : 0010 0110 1000 1011 0011 0000 0000 1100 ;op code=001001;target = 0110 1000 1011 0011 0000 0000 1100. This is j-type
jal 0x68b300c

E: Single Precision Floating Point (Based on section 3.5 in the text)

What decimal number does the bit pattern 0x0C000000 represent if it is a floating-point number? Use the IEEE 754 standard.

Converting to binary(as did in question C)

Sign:

0 - positive

Exponent: $24 - 127 = -103$ (127 is the bias)

Mantissa: 0

Answer in decimal: 1.0×2^{-103}

—
2

F: IEEE 754 Single Precision (Based on section 3.5 in the text)

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. Do not use them. You will be asked questions like this on the exams (using a minifloat format) make sure you know how to do it yourself.

31.25 in binary = 11111.101 = 1.1111101 X 2⁴

Sign: 0 as it is a positive number

Exponent (unadjusted) = 4.

Since, single precision has 32 bits that is 8 exponent bits, therefore :

On adjusting $4 + 2^{8-1} - 1 = 4 + 127 = (131)_{10} = (10000011)_2$

Mantissa(not normalized) = 1.1111101

single precision has 24 fraction bits

Mantissa normalized (removing leading 1 as it is always 1 and making it 23 bits) =

11111010000000000000000

Answer in binary: 0 | 1000 0011 | 111 1101 0000 0000 0000 0000

2

G: IEEE 754 Double Precision (Based on section 3.5 in the text)

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

31.25 in binary = 11111.101 = 1.1111101 X 2⁴

Sign: 0 as it is a positive number

Exponent (unadjusted) = 4.

Since, double precision has 64 bits that is 11 exponent bits, therefore :

On adjusting $4 + 2^{11-1} - 1 = 4 + 1023 = (1027)_{10} = (100\ 0000\ 0011)_2$

Mantissa(not normalized) = 1.1111101

double precision has 53 fraction bits

Mantissa normalized (removing leading 1 as it is always 1 and making it 52 bits) = 111

1101 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0

Answer in binary: 0 | 100 0000 0011 | 1111 1010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

2

H: Sri developed theory question 4 marks.

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. For example, a string value of -705.7481 x 10²⁰ needs to be parsed in some way to interpret it as a floating point value. For this

question, you do not need to show exactly how the characters in the string are converted to numbers, just describe in words each step in the algorithm. For example, a step might include “Read the digits from the left up to the decimal point.”

There is a function to convert a string value into a char array. By this, now our string values are stored in an array of type char where each character of the array is stored as single value in single place of the array. Now using for loop, it is easy to traverse through the array and access only one character of our string value at a time.

Next, we can do a loop, where, as we traverse through the array until character matches “.”. That way we will know at what index our decimal is. Similarly, we can know the indices for where the exponent values start.

Now, we make a variable float x. The way to store the characters as ‘numbers or floating point value’ is, traverse through the array and then use the index number, where the number is stored and do

```
x += array[index] * Math.Pow(10,array.length-1-index)
```

, this way it will be stored as a floating point number. And now since we know where our decimal point is, so after that we can do 10^{-1} , and then power value keeps increasing, so we get correct decimal places.

And when we reach the exponent part (we stored the index where exponent part starts), we can simply, take index+3 (to omit 1,0,^) and then do

```
x * Math.Pow(10,array[index + 3]).
```

This way our string value would have been converted into a floating point value. Yay!