1. [1 mark] Consider the following sequence of bits:

00000000111110

What is the UTF-16 interpretation? Provide the hexcodes and descriptions of the actual character represented.

Solution:

Hexcode: 7E (or 007E or 0x7E), Character: Tilde (~)

*Marking notes: ½ mark for code, ½ mark for character (total 1 mark)*
2. **[6 marks] Floating-Point** (Use IEEE single-precision 32-bit floating point representation)

   a) Convert the decimal number -97.25 to its raw floating-point binary representation (Raw implies that the final answers should consist only of 0s and 1s and no signs or points). Show all steps. **[3.5 marks]**

   Solution:

   -97.25

   Integral part: 97 -&gt; 1100001 (Individual steps for this are not needed) [.5 marks]

   Fractional part: (.25*2 = 0.5  0)
                   (.5 *2 = 1.0  1)

   Therefore, 97.25 in binary is 1100001.01

   Normalized:  1.10000101*2^6  (**"Normalized" word is not necessary**) [.5 marks]

   Therefore, Mantissa (or fraction part) is 10000101000000000000000 [.5 marks]

   Exponent is 6+127 = 133 -&gt; 10000101 [.5 marks]

   With sign bit 1, the raw binary: 1 10000101 10000101000000000000000 [.5 marks]

   b) Add the following two floating point binary numbers using floating point binary addition and give the answer in the normalized binary floating point representation (1.xxxxx.. * 2^y). Also, convert this answer to its decimal value. Show all steps. **[2.5 marks]**

   11000010010110010000000000000000 + 010001000001011000100000000000000

   Solution:

   The normalized binary numbers represented by above are:

   - 1.1011001 *2^5 + 1.001011001 * 2^9  [.5 marks]

   (There may be suffix zeros, but they should be ok as long as the value is correct.)

   Alignment step: - 0.000011011001 *2^9 + 1.0010110001 * 2^9 [.5 marks]

   Perform addition: + 1.00010001001 * 2^9 [.5 marks]

   (The sub-steps for addition performing does not need to be shown, but no problem if it is)

   Conversion to decimal:

   Denormalize: 1000100010.01 [.5 marks]

   Convert the above to decimal: 1*(2^9)+1*(2^5)+1*(2^1)+1*(2^-2) = 546.25 [.5 marks]

   (The step needs to be shown. Some answers may include zero bits as well, which is ok)
3. [4 marks] MIPS

a) Convert the C code below into exactly 3 MIPS instructions. Assume the base address of Z is stored in $17, the base address of K is stored in $18. Use $8 as the only temporary register. (No other temporary registers are allowed) [2 marks]

\[ Z[7] = -7 + K[0]; \]

Solution:

lw $8, 0($18) \hspace{1cm} [0.5 mark]
addi $8, $8, -7 \hspace{1cm} [1 mark]
sw $8, 28($17) \hspace{1cm} [0.5 mark]

For questions 3(b),

- Your programs must compile with the given MIPS assembler `binasm` in the student.cs environment. No partial marks if the program does not execute correctly.
- Your solution should include comments at the beginning that include your name, your Quest ID, a brief description of the program, and a list of the registers with a brief description of how they are being used.
- Remember when you are checking your results that the values that appear in the registers are all hexadecimal numbers.

b) Write a MIPS program to calculate the result of the following expression and place the result in register $7. You can any registers between $8-$15 as temporary registers. [2 marks]

\[ (30987 - 20790)^2 \]

Solution:

(The answer in $7$ should be: 6329739 – hex value. Below is a simple approach. There may be other approaches to solve the same problem. Zero marks if the program does not execute correctly.)

addi $10, $0, 30987 \hspace{1cm} [0.5 marks]
addi $10, $10, -20790 \hspace{1cm} [0.5 marks]
mult $10, $10 \hspace{1cm} [0.5 marks]
mflo $7 \hspace{1cm} [0.5 marks]
jr $31