

CSE 238/2038/2138 Systems Programming

Project 1

Binary Data Converter

Due: 02.12.2020 11:59PM

The purpose of this assignment is to become more familiar with bit-level representations of integers and floating-point numbers.

In this project, you will implement an application, in C, Java or Python programming language; that takes

- a hexadecimal number and
- the data type to be converted,

as input and converts the number according to the predefined format and gives the converted data as output.

The data type can be any of the following:

- signed integer (S)
- unsigned integer (U)
- floating point number (F)

The size of the input can be 1, 2, 3 and 4 bytes, if input is other than these sizes then an error message will be given.

- If the selected data type is signed integer, your program will convert the number using 2's complement representation.
- If the selected data type is unsigned integer, number will be converted using unsigned integer representation.
- If the selected data type is floating point number, you will use IEEE-like format. The number of exponent bits according to given data size will be like the following:
 - if 1 byte (i.e., 8 bits), 4 bits will be used for exponent part
 - if 2 bytes (i.e., 16 bits), 6 bits will be used for exponent part
 - if 3 bytes (i.e., 24 bits), 8 bits will be used for exponent part
 - if 4 bytes (i.e., 32 bits), 10 bits will be used for exponent part
 - For each given data size 1 bit will be used for sign and remaining bits will be used for fraction.
 - While calculating the mantissa to get the floating-point value, you will only use the first 13 bits of the fraction part (If the data size is 3 or 4 bytes). You will use "round to even" method for rounding fraction bits to 13 bits.

Details about the program are listed below:

- At the beginning of the execution, your program will prompt for the input. If the input has odd number of characters complete it to even by adding 0 to the MSB.
 - For example;
 - If the number entered is A57, it will be stored as 0A57 and its size is 2 bytes.
 - If the number is 12345, it will be stored as 012345, and its size is 3 bytes.
 - If the input is 123456789, then an error message is given. (bigger than 4 byte)
 - Any inappropriate input results an error message. (Only digits and characters from A to F will be accepted as an input)
- After a valid input is taken, the user will be prompted for the data type:

Example:

Data type: F

- And then your program will calculate the decimal value of input with the given information:

Example 1:

Enter the number: 400190F0

Data type: F

- Floating point number is:
40 01 90 f0, in binary: 0100 0000 0000 0001 1001 0000 1111 0000
- In the specification, we are given that our 4 byte IEEE-like floating point numbers have 10 bits of exponent part, so Bias = $2^{10-1} - 1 = 511$:
Sign bit = 0
Exponent = $(1000000000)_2 = 512$
Fraction = 000011001000011110000 → rounded fraction = 0000110010001
mantissa = $1 + 1/32 + 1/64 + 1/512 + 1/8192 = 1.0489501953125$
Decimal value = $(-1)^0 * 1.0489501953125 * 2^{512-511} = 2.097900390625$
The printed value will be: 2.09790

Example 2:

- If the input;
Enter the number: 80180000
Data type: F
- Floating point number is:
80 18 00 00, in binary: 1000 0000 0001 1000 0000 0000 0000 0000
- In the specification, we are given that our 4 byte IEEE-like floating point numbers have 10 bits of exponent part, so Bias = $2^{10-1} - 1 = 511$:
Sign bit = 1
Exponent = 0000000000 → denormalized number
Fraction = 11000000000000000000 → rounded fraction = 11000000000000000000

mantissa = $0 + 1/2 + 1/4 = 0.75$
Decimal value = $(-1)^1 * 0.75 * 2^{1-511} = -2.23750222e-154$
The printed value will be: **-2.23750e-154**

Example 3:

Enter the number: **83A**

Data type: **S**

- Floating point number is:
08 3A, in binary: **0000 1000 0011 1010**
- It is positive binary number and it will be directly converted to decimal
The printed value will be : **2106**

- Floating point numbers may be NaN, +0, -0 or infinity. In these cases, the output will be:
 - NaN
 - ∞
 - $-\infty$
 - 0
 - -0
- For the output, the floating-point numbers will have precision of maximum 5 digits after the decimal point.
- **You cannot use library functions** for the binary to decimal conversions.

Your grade will be computed out of a maximum of **80** points.

For the remaining **20** points, you will have a project quiz. The date of this quiz will be announced later.

You will submit the source code of your program using the Canvas service. The file name should include your name and surname, e.g., *fatmacorutergin.c*

This is an individual project.