

UMass Boston CS 240
Homework 2
Due 02/25/2020 11:00 AM

1 Notes & Requirements

1. Way of submission will be announced later.
2. Write your full name and Unix name in your code comments.
3. No late homework.
4. For floats decimal to binary conversion, read the bits from memory is not acceptable, which will get 0 points. You could use that for testing
5. You have to implement the algorithms described in this instruction.
6. Very good reading reference.

<https://indepth.dev/the-simple-math-behind-decimal-binary-conversion-algorithms/>

2 Integer Decimal to Base-q Conversion

Complete the function `void num2q(int num, char num_q[], int q)` in homework2, which converts an integer number to a base-q number string.

- Use letter for digits that bigger than 9, such that A for 10, B for 11 etc.
- To make things simple, we only use capital letter here.
- The integer to be tested here will be smaller than maximum integer.
- The q to be tested will be bigger than 2 and less than 20 and you do not have to consider the exceptions that if q is too big that there are not enough letters for representing digits.

3 Integer Base-q Conversion to Decimal

Do the reverse of above, complete the function `int q2num(char num_q[], int q)` in homework2, which converts a base-q number string to a integer number.

4 Floats Decimal to Binary Conversion

Floats have two parts, i.e. integer part and fraction part, and for each of these two parts they may be represented in decimal format or in binary format, like below:

8.125	decimal format
100.001	binary format

There are rules for converting them interchangeably and in this homework you are required to implement what has been described below.

4.1 Decimal Integer to Binary

To convert integer to binary, start with the integer in question and divide it by 2 keeping notice of the quotient and the remainder. Continue dividing the quotient by 2 until you get a quotient of zero. Then just write out the remainders in the reverse order.

Here is an example of such conversion using the integer 12.

First, let's divide the number by two specifying quotient and remainder:

$$\text{Dividend(Dd)} = \text{Divisor(Dr)} * \text{Quotient(Q)} + \text{Reminder(R)}$$

Dd	Dr	Q	R
12 =	2 *	6 +	0
6 =	2 *	3 +	0
3 =	2 *	1 +	1
1 =	2 *	0 +	1

Now, we simply need to write out the remainder in the reverse order **1100** . So, 12 in decimal system is represented as **1100** in binary.

4.2 Decimal Fraction to Binary

To convert fraction to binary, start with the fraction in question and multiply it by 2 keeping notice of the resulting integer and fractional part. Continue multiplying by 2 until you get a resulting fractional part equal to zero. Then just write out the integer parts from the results of each multiplication.

Here is an example of such conversion using the fraction 0.375.

$$\begin{array}{l} \text{Multiplicand(Mt)} * \text{Multiplier(Mr)} = \text{Product} \\ \text{(Fraction(F) + Integer(I))} \end{array}$$

Mt	Mr	F	I
0.375 *	2 =	0.75 +	0
0.75 *	2 =	0.5 +	1
0.5 *	2 =	0 +	1

Now, let's just write out the resulting integer part at each step **0.011**. So, 0.375 in decimal system is represented as **0.011** in binary.

4.3 Floats to binary string

Suppose we let a binary string to represent a float. And this binary string is delimited by the character " . " into two parts – integer part and fraction part, like discussed above.

Here are the requirements of this kind of binary string here:

1. For this string, it has to have " . " in it.
2. For the fraction, it at least contains 1 bit and at most 8 bits
3. For the 8 bits fraction, you don't need to use rounding.
4. For the integer part, it at least contains 1 bit.

Based on these requirements, complete the function **void f2b(float f, char f_b[])** in homework2 to convert a float number into a binary string.

4.4 Notes

1. I will testify the floats within the range $[2^{-8}, 2^8]$ on the function of **void f2b(float f, char f_b[])**

5 Floats Binary to Decimal Conversion

Do the reverse of above, complete the function **float b2f(char f_b[])** in homework2, such that a binary string being converted to a float.

6 Bonus question: Single-precision floating-point format

If you could convert the floats into correct single-precision floating-point format string, you will get bonus points up to 50.

The requirements:

1. The function name should be "void f2sp(float f, char f_sp[])" in the "f2sp.c" file
2. For testing, I will read the bits from memory like we did in class and compare the char array with that coming from the function that you write.
3. The number range to be tested will be $[2^{-149}, 2^{128}]$.
4. Virtually you should just pass the loop of assertions in the test.c file
5. You could come up with your own testing for some part functionality that you write. In this case you should comment it clearly and it has to be relevant.
6. Write your code within the bonus folder.