PART 2

First we need convert decimal 0.0098 to binary. So we will do 2 multiplication 0.0098 several times.

Every multiplication we write 0 or 1. If we get the integer result of this operation we will write 1 after result will minus one. Example;

|  |  |
| --- | --- |
| Operation | Binary |
| *0.0098x2=0,0196* | 0 |
| *0.0098x2=0,0392* | 00 |
| *0.0098x2=0,0784* | 000 |
| *0.0098x2=0,1568* | 0000 |
| *0.0098x2=0,3136* | 00000 |
| *0.0098x2=0,6272* | 000000 |
| *0.0098x2=1,2544*  *1,2544-1=0,2544* | 0000001 |
| *0,2544x2=0,5088* | 00000010 |
| *0,5088x2=1,0176*  *1,0176-1=0,0176* | 000000101 |
| *0,0176\*2=0,0352* | 0000001010 |
| *0,0352\*2=0,0704* | 00000010100 |
| *.* | . |
| *.* | . |
| *.* | . |
| *.* | . |
| *.* | . |
| *0,5008x2=1,0016*  *1,0016-1=0,0016* | 000000101000001001000000101101112 |

Now we have 32-bit binary reprasantation of 0.0098 Decimal Value.

We want to convert 32-bit value to IEEE 32 bits floating point format.

IEEE 32 bits floating point format:

* İf number is positive sign is 0. But number is negative sign is 1.And sign have a 1-bit
* İn this format, Exponent substracting 2k-1-1.(k is bit of exponent).And exponent have a 8-bit.So exponent substracting 28-1-1 = 127.
* Mantissa have 23-bits.This is a significant portion.

Now we will convert 0.0098 to IEEE 32 bits floating point format. We converted 0.0098 decimal to 00000010100000100100000010110111 binary.

And 0.0098 is positive so this sign is 0.So now ,we need found exponent and mantissa

We will convert 000000101000001001000000101101112 to IEEE 32 bits floating point format. We need organize this binary number.

|  |
| --- |
| Operations |
| *0,00000010100000100100000010110111/2=0,0000010100000100100000010110111x2-1* |
| *0,0000010100000100100000010110111/2=0,000010100000100100000010110111x2-2* |
| *0,000010100000100100000010110111/2=0,00010100000100100000010110111x2-3* |
| *0,00010100000100100000010110111/2=0,0010100000100100000010110111x2-4* |
| *0,0010100000100100000010110111/2=0,010100000100100000010110111x2-5* |
| *0,010100000100100000010110111/2=0,10100000100100000010110111x2-6* |
| *0,10100000100100000010110111/2=1,0100000100100000010110111x2-7* |

After organized 0,00000010100000100100000010110111, we have a 1,0100000100100000010110111x2-7.

We will take 2-7 for exponent and 127+(-7)=120. Now we convert 120 to binary.

12010 = 011110002

We have achieved exponent. And it is 011110002.

Now we are take resiude of the number(1,01000001001000000101101112). We cut off it before the comma.

1,01000001001000000101101112 >>> 01000001001000000101101112

Now we have achieved mantissa. And it is 01000001001000000101101112.

After that. We also have achieved IEEE 32-bits floating format:

**0.0098 Decimal to IEEE 32-bits Floating Format**

2)

We will convert 110000010111000000000000000000002 binary to decimal.

First we will separate this binary to the parts.



Sign is 1 so number is negative.

Exponent is 100000102. We need convert this binary to decimal.

100000102 = 13010

Mantissa multiplicationed with 2 unknown times. But we need know how many times mantissa multiplicationed with 2.So we will minus decimal value of exponent to 127.And we find k on 2k. So;

130-127=3 & 3=k, result is 23

Normally if we binary to IEEE 32-bits floating format.We separated of before comma.So now we adding back before of comma.

11100000000000000000000 >>> 1,11100000000000000000000x23

After if we did this multiplication. We will have this binary:

1111000000000000000000002

And now we can convert this binary to decimal number.

11112 = 1510

And last our sign is 1. So this number is negative.

-1510

Result is;

110000010111000000000000000000002 >>> -1510

PART 3

BubbleSort

* **STEP 1.** Take the first two numbers.
* **STEP 2.** Compare the numbers.
* **STEP 3.** Store the large number in temporary location
* **STEP 4.** Write small number instead of bigger number
* **STEP 5**. Write number in temporary location instead of small number.
* **STEP 6.** İf sequence is end. Write the number in temporary location to end of sequence. And HALT
* **STEP 7.** İf sequence isn’t end. Take new number.
* **STEP 8.** G oto the **STEP 2.**