HW1-Fundamentals

0.2 Binary Numbers

Exercise 3

Convert the following base 10 numbers to binary. Use overbar notation for nonterminating binary numbers. (a) 10.5 (b) 1/3 (c) 5/7 (d) 12.8 (e) 55.4 (f) 0.1

a)

$$(10.5)_{10} = (1010.1)_2$$

Integer part	Fractional part
10/2 = 5.0→0 5/2=2.5→1 2/2=1.0→0 1/2=0.5→1	0.5*2=1.0 0.0*2=0.0
1010	1

b)

$$(1/3)_{10} = (0.\overline{3})_{10} = (0.\overline{01})_2$$

Integer part	Fractional part
0/2 = 0→0	$0.\overline{3}*2=0.\overline{6}$ $0.\overline{6}*2=1.\overline{3}$ $0.\overline{3}*2=0.\overline{6}$ ….
0	01 01 01….

c)

$$(5/7)_{10} = \left(0.\overline{714285}\right)_{10} = (0.\overline{101})_2$$

Integer part	Fractional part
0/2 = 0→0	$0.\overline{714285}*2=1.\overline{428571}$ $0..\overline{428571}*2=0.\overline{857142}$ $0.\overline{857142}*2=1.\overline{714285}$ $0.\overline{714285}*2=1.\overline{428571}$ …
0	101 101 101…

d)

$$(12.8)_{10} = (0.\overline{1100})_2$$

Integer part	Fractional part

Alejandro C. Parra Garcia

12/2 = 6.0 → 0 6/2 = 3.0 → 0 3/2 = 1.5 → 1 1/2=0.5 → 1	0.8*2=1.6 0.6*2=1.2 0.2*2=0.4 0.4*2=0.8 0.8*2=1.6 …
1100	1100 1100 1100 …

e)

$$(55.4)_{10} = (110111.\overline{0110})_2$$

Integer part	Fractional part
55/2 = 27.5 → 1 27/2 = 13.5 → 1 13/2 = 6.5 → 1 6/2=3.0 → 0 3/2 = 1.5 → 1 1/2=0.5 → 1	0.4*2=0.8 0.8*2=1.6 0.6*2=1.2 0.2*2=0.4 0.4*2=0.8 …
110111	0110 0110 0110 …

f)

$$(0.1)_{10} = (0.0\overline{0011})_2$$

Integer part	Fractional part
0/2 = 0.0 → 0	0.1*2=0.2 0.2*2=0.4 0.4*2=0.8 0.8*2=1.6 0.6*2=1.2 0.2*2=0.4 …
0	0 0011 0011 0011 …

Exercise 7

Convert the following binary numbers to base 10: (a) 1010101 (b) 1011.101 (c) 10111. $\overline{01}$ (d) 110. $\overline{10}$ (e) 10. $\overline{110}$ (f) 110.1$\overline{101}$ (g) 10.010$\overline{1101}$ (h) 111.$\overline{1}$

a)

$$(1010101)_2 = 1 * 2^6 + 0 * 2^5 + 1 * 2^4 + 0 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0$$

$$64 + 0 + 16 + 0 + 4 + 0 + 1 = (85)_{10}$$

b)

$$(1011.101\ )_2 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 + 1 * 2^{-1} + 0 * 2^{-2} + 1 * 2^{-3}$$

$$8 + 0 + 2 + 1 + \frac{1}{2} + 0 + \frac{1}{8} = (11.625)_{10}$$

c)

$$(10111.\overline{01})_2$$

Integer Part:

$$(10111)_2 = 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 1*2^0 = 16 + 4 + 2 + 1 = 23$$

Fractional part:

$$(0.\overline{01})_2$$

$$(0.\overline{01})_2 * (2^2)_{10} = (01.\overline{01})_2 => x*4 = 1 + x => x = 1/3$$

$$(10111.\overline{01})_2 = (23.\overline{3})_{10} = \left(\frac{70}{3}\right)_{10}$$

d)

$$(110.\overline{10})_2$$

Integer Part:

$$(110)_2 = 1*2^2 + 1*2^1 + 0*2^0 = 4 + 2 = 6$$

Fractional part:

$$(0.\overline{10})_2$$

$$(0.\overline{10})_2 * (2^2)_{10} = (10.\overline{10})_2 => x*4 = 2 + x => x = 2/3$$

$$(110.\overline{10})_2 = (6.\overline{6})_{10} = \left(\frac{20}{3}\right)_{10}$$

e)

$$(10.\overline{110})_2$$

Integer Part:

$$(10)_2 = 1*2^1 + 0*2^0 = 2$$

Fractional part:

$$(0.\overline{110})_2$$

$$(0.\overline{110})_2 * (2^3)_{10} = (110.\overline{110})_2 => x*8 = 6 + x => x = 6/7$$

$$(10.\overline{110})_2 = \left(2.\overline{857142}\right)_{10} = \left(\frac{20}{7}\right)_{10}$$

f)

$$(110.1\overline{101})_2$$

Integer Part:

$$(110)_2 = 1*2^2 + 1*2^1 + 0*2^0 = 4 + 2 = 6$$

Fractional part:

$$(0.1\overline{101})_2$$

$$(0.1\overline{101})_2 * (2^1)_{10} = (1.\overline{101})_2 => x*2 = 1 + (0.\overline{101})_2$$

$$(0.\overline{101})_2 * (2^3)_{10} = (101.\overline{101})_2 => y * 8 = 5 + y => y = 5/7$$

$$x * 2 = 1 + (0.\overline{101})_2 => x * 2 = 1 + \frac{5}{7} => x = 6/7$$

$$(110.1\overline{101})_2 = (6.\overline{857142})_{10} = \left(\frac{48}{7}\right)_{10}$$

g)

$$(10.010\overline{1101})_2$$

Integer Part:

$$(10)_2 = 1 * 2^1 + 0 * 2^0 = 2$$

Fractional part:

$$(0.010\overline{1101})_2$$

$$(0.010\overline{1101})_2 * (2^3)_{10} = (010.\overline{1101})_2 => x * 8 = 2 + (0.\overline{1101})_2$$

$$(0.\overline{1101})_2 * (2^4)_{10} = (1101.\overline{1101})_2 => y * 16 = 13 + y => y = 13/15$$

$$x * 8 = 2 + (0.\overline{1101})_2 => x * 8 = 2 + \frac{13}{15} => x = 43/120$$

$$(10.010\overline{1101})_2 = (2.358\overline{3})_{10} = \left(\frac{283}{120}\right)_{10}$$

h)

$$(111.\overline{1})_2$$

Integer Part:

$$(111)_2 = 1 * 2^2 + 1 * 2^1 + 1 * 2^0 = 4 + 2 + 1 = 7$$

Fractional part:

$$(0.\overline{1})_2$$

$$(0.\overline{1})_2 * (2^1)_{10} = (1.\overline{1})_2 => x * 2 = 1 + x => x = 1$$

$$(111.\overline{1})_2 = (7 + 1)_{10} = (8)_{10}$$

0.3 FP Representation of Real Numbers

Exercise 4

Find the largest integer k for which fl(19 + 2^{-k}) > fl(19) in double precision floating point arithmetic.

$$(19)_{10} = (10011)_2$$

$$fl(19) = 1.00110 \ldots 0 * 2^4$$

$$\left(2^{-\mathrm{k}}\right)_{10} = (0.0 \ldots 01\langle k\ decimal\ digits\rangle)_2$$

$$fl\left(2^{-\mathrm{k}}\right) = 1.0 \ldots 0 * 2^{-k}$$

$$fl\left(19\ +\ 2^{-\mathrm{k}}\ \right) = 1.00110 \ldots 0 * 2^4 + 1.0 \ldots 0 * 2^{-k}$$

As we have calculated the floating point of 19, is 1.00110…0*2^4, so the next number above fl(19) is:

$$fl(19) = 1.00110 \ldots 0\langle 52\ decimal\ digits\rangle * 2^4 < 1.00110 \ldots 01\langle 52\ decimal\ digits\rangle * 2^4$$

Lets call this number fl(19+) indicating that it is the number immediately superior of fl(19)

$$fl\left(19\ +\ 2^{-\mathrm{k}}\ \right) = fl(19+)$$

$$1.00110 \ldots 0 * 2^4 + 1.0 \ldots 0 * 2^{-k} = 1.00110 \ldots 01\langle 52\ decimal\ digits\rangle * 2^4$$

$$1.0 \ldots 0 * 2^{-k} = 0.0 \ldots 01\langle 52\ decimal\ digits\rangle * 2^4$$

$$1.0 \ldots 0 * 2^{-k} = 1.0 \ldots 0 * 2^{4-52}$$

$$-k = 4 - 52 => k = 48$$

Let’s check:

fl(19 + 2^{-48})

$$fl(19) = 1.00110 \ldots 0 * 2^4$$

$$(2^{-48})_{10} = (0.0 \ldots 01\langle 48\ decimal\ digits\rangle)_2$$

$$fl(2^{-48}) = 1.0 \ldots 0 * 2^{-48}$$

Now we align the decimal points,

$$1.0 \ldots 0 * 2^{-48} = 0.0 \ldots 01\langle 52\ decimal\ digits\rangle * 2^4$$

Now we add:

1	.	0	0	1	1	0	…	0	*2^4
0	.	0	0	0	0	0	…	1	*2^4
1	.	0	0	1	1	0	…	1	*2^4

Result:

$$fl\left(19\ +\ 2^{-\mathrm{k}}\ \right) = 1.00110 \ldots 01\langle 52\ decimal\ digits\rangle * 2^4$$

This number is the equal to

$$fl(19+) = 1.00110 \ldots 01\langle 52\ decimal\ digits\rangle * 2^4$$

And thus K = 48

Exercise 8

Is 1/3 + 2/3 exactly equal to 1 in double precision floating point arithmetic, using the IEEE Rounding to Nearest Rule? You will need to use fl(1/3) and fl(2/3) from Exercise 1. Does this help explain why the rule is expressed as it is? Would the sum be the same if chopping after bit 52 were used instead of IEEE rounding?

$$(1/3)_{10} = (\,0.\overline{01})_2$$

$$fl(1/3) = 1.010101 \ldots 01\langle 52\ decimal\ digits\rangle 0101 * 2^{-2}$$

Rounding:

$$fl(1/3) = 1.010101 \ldots 01\langle 52\ decimal\ digits\rangle * 2^{-2}$$

$$(2/3)_{10} = (\,0.\overline{10})_2$$

$$fl(2/3) = 1.01010 \ldots 01\langle 52\ decimal\ digits\rangle 0101 * 2^{-1}$$

Rounding:

$$fl(2/3) = 1.01010 \ldots 01\langle 52\ decimal\ digits\rangle * 2^{-1}$$

Let's add:

$$fl(1/3) + fl(2/3) = 1.0101 \ldots 01 * 2^{-2} + 1.0101 \ldots 01 * 2^{-1}$$

First, we need to align the decimal point:

$$fl(1/3) = 1.0101 \ldots 01 * 2^{-2} = 0.10101 \ldots 01\langle 53\ decimal\ digits\rangle * 2^{-1}$$

Second, we add:

0	.	1	0	1	0	…	0	1	0	1	$*2^{-1}$
1	.	0	1	0	1	…	1	0	1		$*2^{-1}$
1	.	1	1	1	1	…	1	1	1	1	$*2^{-1}$

Result:

$$fl(1/3) + fl(2/3) = 1.11 \ldots 11\langle 53\ decimal\ digits\rangle * 2^{-1}$$

Rounding:

$$1.11 \ldots 1\langle 52\ decimal\ digits\rangle 1 * 2^{-1} = 10.0 \ldots 0 * 2^{-1}$$

$$10.0 \ldots 0 * 2^{-1} = 1.0 \ldots 0 * 2^{0}$$

$$1.0 \ldots 0 * 2^{0} = fl(1)$$

The sum is exactly 1. This is due to the rounding error of the resulting addition cancelling the rounding errors from the numbers.

If we would chop after bit 52, the sum will we the same, but when we round the sum, it will be chopped meaning the results won't we 1:

Alejandro C. Parra Garcia

$$fl(1/3) + fl(2/3) = 1.11 \ldots 1\langle 52\ decimal\ digits\rangle * 2^{-1} = 0.11 \ldots 111 * 2^{0} \neq fl(1)$$

Exercise 11

Does the associative law hold for IEEE computer addition?

No, it doesn't. That is because in the in-between steps we are rounding the numbers, this can lead to a different result depending on the order of operations we perform with the numbers.

0.4 Loss of Significance

Exercise 1

a)

$$\frac{1 - \sec x}{\tan^2 x}$$

Let's rewrite it as:

$$\frac{1 - \sec x}{\tan^2 x} = \frac{(1 - \sec x)(1 + \sec x)}{\tan^2 x * (1 + \sec x)} = \frac{1 + \sec x - \sec x - \sec^2 x}{\tan^2 x * (1 + \sec x)} = \frac{1 - \sec^2 x}{\tan^2 x * (1 + \sec x)}$$

$$\frac{1 - \sec^2 x}{\tan^2 x * (1 + \sec x)} = \frac{1 - \frac{1}{\cos^2(x)}}{\tan^2 x * (1 + \sec x)} = \frac{\frac{\cos^2(x) - 1}{\cos^2(x)}}{\tan^2 x * (1 + \sec x)} = \frac{\frac{-\sin^2(x)}{\cos^2(x)}}{\tan^2 x * (1 + \sec x)}$$

$$\frac{-\left(\frac{\sin(x)}{\cos(x)}\right)^2}{\tan^2 x * (1 + \sec x)} = \frac{-\tan^2(x)}{\tan^2 x * (1 + \sec x)} = \frac{-1}{(1 + \sec x)}$$

$$\frac{1 - \sec x}{\tan^2 x} = \frac{-1}{(1 + \sec x)}$$

Now let's print the results.

First column the value of X, Second column the value of the original equation, Third the new equation

```
0.1     [mpf('-0.49874791371143462'), [mpf('-0.49874791371142879'),
0.01     mpf('-0.49998749979095553'),  mpf('-0.49998749979166379'),
0.001    mpf('-0.49999987501428939'),  mpf('-0.49999987499997917'),
0.0001   mpf('-0.4999999362793118'),   mpf('-0.4999999875000001'),
1e-05    mpf('-0.50000004133685205'),  mpf('-0.499999999875'),
1e-06    mpf('-0.50004445029083722'),  mpf('-0.49999999999987499'),
1e-07    mpf('-0.51070259132756868'),  mpf('-0.49999999999999867'),
1e-08    mpf('0.0'),                   mpf('-0.5'),
1e-09    mpf('0.0'),                   mpf('-0.5'),
1e-10    mpf('0.0'),                   mpf('-0.5'),
1e-11    mpf('0.0'),                   mpf('-0.5'),
1e-12    mpf('0.0'),                   mpf('-0.5'),
1e-13    mpf('0.0'),                   mpf('-0.5'),
1e-14    mpf('0.0')]                   mpf('-0.5')]
```

b)

$$\frac{1-(1-x)^3}{x}$$

Let’s rewrite it as:

$$(1-x)^3 = (1-x)(1-x)(1-x) = (1-x)*(1-2x+x^2) =$$

$$1-2x+x^2-(x-2x^2+x^3) = 1-2x+x^2-x+2x^2-x^3$$

$$1-2x+x^2-x+2x^2-x^3 = 1-3x+3x^2-x^3$$

$$\frac{1-(1-x)^3}{x} = \frac{1-(1-3x+3x^2-x^3)}{x} = \frac{1-1+3x-3x^2+x^3}{x}$$

$$\frac{1-1+3x-3x^2+x^3}{x} = \frac{3x-3x^2+x^3}{x} = \frac{x*(3-3x+x^2)}{x} = (3-3x+x^2)$$

$$\frac{1-(1-x)^3}{x} = (3-3x+x^2)$$

Now let’s print the results.

First column the value of X, Second column the value of the original equation, Third the new equation

0.1	[2.7099999999999999,	[2.71,
0.01	2.9700999999999977,	2.9701000000000004,
0.001	2.9970009999999999,	2.997001,
0.0001	2.9997000100001614,	2.9997000099999998,
1e-05	2.9999700000837843,	2.9999700001,
1e-06	2.99999700004161,	2.999997000001,
1e-07	2.999999698660716,	2.9999997000000103,
1e-08	2.999999981767587,	2.99999997,
1e-09	2.9999999151542056,	2.999999997,
1e-10	3.000000248221113,	2.9999999997,
1e-11	3.000000248221113,	2.99999999997,
1e-12	2.9999336348396355,	2.999999999997,
1e-13	3.000932835561798,	2.9999999999997,
1e-14	2.9976021664879227]	2.99999999999997]

Exercise 2

Find the smallest value of p for which the expression calculated in double precision arithmetic at x = 10−p has no correct significant digits. (Hint: First find the limit of the expression as x → 0.)

a)

$$\frac{\tan(x) - x}{x^3}$$

```
0.1       [mpf('0.33467208545054355'),
0.01       mpf('0.33334666720702394'),
0.001      mpf('0.33333346673158903'),
0.0001     mpf('0.33333333651890806'),
1e-05      mpf('0.3333328757329847'),
1e-06      mpf('0.33330746474456724'),
1e-07      mpf('0.3308722450212111'),
1e-08      mpf('0.0'),
1e-09      mpf('0.0')]
```

The smallest value of p for which the expression calculated in double precision arithmetic at x = 10−p has no correct significant digits is for p=8

b)

$$\frac{e^x + \cos(x) - \sin(x) - 2}{x^3}$$

```
0.1          [mpf('0.000341666706845433777'),
0.01          mpf('3.3416666678220963e-7'),
0.001         mpf('3.3341684968490881e-10'),
0.0001        mpf('3.3351099659739702e-13'),
1e-05         mpf('4.4408920985006262e-16'),
1e-06         mpf('0.0'),
1e-07         mpf('-2.2204460492503131e-16'),
1e-08         mpf('0.0'),
1e-09         mpf('0.0'),
1e-10         mpf('0.0'),
1e-11         mpf('0.0'),
1e-12         mpf('0.0'),
1e-13         mpf('0.0')]
```

The smallest value of p for which the expression calculated in double precision arithmetic at x = 10−p has no correct significant digits is for p=6

Exercise 3

Evaluate the quantity $a + \sqrt{a^2 + b^2}$ to four correct significant digits, where a = −12345678987654321 and b = 123

For this part I'm going to use the library decimal, in order to get a higher decimal precision.

```
from decimal import Decimal
from decimal import *
getcontext().prec = 33

a=Decimal(-12345678987654321)
b=Decimal(123)
```

We indicate that the numbers are in decimal type with a precision of 33. That way we don't lose the decimal values of the square root:

```
[128] a**2

      Decimal('152415789666209420210333789971041')

[129] b**2

      Decimal('15129')

[130] a**2+b**2

      Decimal('152415789666209420210333789986170')

[131] (a**2+b**2)**(Decimal(1/2))

      Decimal('12345678987654321.0000000000006127')

      a+(a**2+b**2)**(Decimal(1/2))

      Decimal('6.127E-13')
```

We get that the result is $6.127*10^{-13}$

If we where to use normal double precision, we will get this other result:

```
a = float(-12345678987654321)
b = float(123)

solution = a + math.sqrt((a**2)+(b**2))
f'{solution:.4}'

'0.0'
```

This is because the value of 'a' for example get store, but it loses the least significant digit:

```
f'{a:.33}'

'-12345678987654320.0'
```

So, in return every decimal number will get lost, and in return the square root is going to have the exact same value of the 'a'

Code

```
from mpmath import *
import math
```

Exercise 1

part a)

```
result1=[]
result2=[]
for i in range(1,15):
  value=10**(-i)
  print(value)

  vv1=(1-sec(value))/(tan(value)**2)
  result1.append(vv1)

  vv2=(-1)/(1+sec(value))
  result2.append(vv2)
```

```
0.1
0.01
0.001
0.0001
1e-05
1e-06
1e-07
1e-08
1e-09
1e-10
1e-11
1e-12
1e-13
1e-14
```

```
[3]  result1

[mpf('-0.49874791371143462'),
 mpf('-0.49998749979095553'),
 mpf('-0.49999987501428939'),
 mpf('-0.49999999362793118'),
 mpf('-0.50000004133685205'),
 mpf('-0.50004445029083722'),
 mpf('-0.51070259132756868'),
 mpf('0.0'),
 mpf('0.0'),
 mpf('0.0'),
 mpf('0.0'),
 mpf('0.0'),
 mpf('0.0'),
 mpf('0.0')]
```

```
[4]  result2

[mpf('-0.49874791371142879'),
 mpf('-0.49998749979166379'),
 mpf('-0.49999987499997917'),
 mpf('-0.49999999875000001'),
 mpf('-0.4999999999875'),
 mpf('-0.49999999999987499'),
 mpf('-0.49999999999999867'),
 mpf('-0.5'),
 mpf('-0.5'),
 mpf('-0.5'),
 mpf('-0.5'),
 mpf('-0.5'),
 mpf('-0.5'),
 mpf('-0.5')]
```

part b)

```
[7] result1=[]
    result2=[]
    for i in range(1,15):
      value=10**(-i)
      print(value)

      vv1=(1-(1-value)**3)/(value)
      result1.append(vv1)

      vv2=(3-3*value+value**2)
      result2.append(vv2)
```

```
0.1
0.01
0.001
0.0001
1e-05
1e-06
1e-07
1e-08
1e-09
1e-10
1e-11
1e-12
1e-13
1e-14
```


Alejandro C. Parra Garcia

```
[8] result1
```

```
[2.709999999999999,
 2.9700999999999977,
 2.997000999999999,
 2.9997000100001614,
 2.9999700000837843,
 2.99999700004161,
 2.999999698660716,
 2.999999981767587,
 2.9999999151542056,
 3.000000248221113,
 3.000000248221113,
 2.9999336348396355,
 3.000932835561798,
 2.9976021664879227]
```

```
[9] result2
```

```
[2.71,
 2.9701000000000004,
 2.997001,
 2.9997000099999998,
 2.9999700001,
 2.999997000001,
 2.9999997000000103,
 2.99999997,
 2.999999997,
 2.9999999997,
 2.99999999997,
 2.999999999997,
 2.9999999999997,
 2.99999999999997]
```

Exercise 2

part a)

```
[21] solution=[]
     for i in range(1,10):
       x=10**(-i)
       print(x)

       sol=(tan(x)-x)/(x**3)
       solution.append(sol)
```

```
0.1
0.01
0.001
0.0001
1e-05
1e-06
1e-07
1e-08
1e-09
```

```
solution
```

```
[mpf('0.334672085450544355'),
 mpf('0.333346667207020394'),
 mpf('0.333333466731589003'),
 mpf('0.333333336518908006'),
 mpf('0.333332875732984 7'),
 mpf('0.333307464744567 24'),
 mpf('0.330872245021211 1'),
 mpf('0.0'),
 mpf('0.0')]
```

part b)

```
solution=[]
for i in range(1,14):
  x=10**(-i)
  print(x)

  sol=(math.exp(x)+cos(x)-sin(x)-2)
  solution.append(sol)
```

```
0.1
0.01
0.001
0.0001
1e-05
1e-06
1e-07
1e-08
1e-09
1e-10
1e-11
1e-12
1e-13
```

```
solution
```

```
[mpf('0.000341666706845433377'),
 mpf('3.34166666678220963e-7'),
 mpf('3.33416849684908881e-10'),
 mpf('3.33510996597397702e-13'),
 mpf('4.44089209850062626e-16'),
 mpf('0.0'),
 mpf('-2.22044604925031308e-16'),
 mpf('0.0'),
 mpf('0.0'),
 mpf('0.0'),
 mpf('0.0'),
 mpf('0.0'),
 mpf('0.0')]
```

Exercise 3

```
[121] from decimal import Decimal
      from decimal import *
      getcontext().prec = 33

      a=Decimal(-12345678987654321)
      b=Decimal(123)
```

```
[128] a**2
```

```
Decimal('152415789666209420210333789971041')
```

```
[129] b**2
```

```
Decimal('15129')
```

```
[130] a**2+b**2
```

```
Decimal('152415789666209420210333789986170')
```

```
[131] (a**2+b**2)**(Decimal(1/2))
```

```
Decimal('12345678987654321.0000000000006127')
```

```
[132] a+(a**2+b**2)**(Decimal(1/2))
```

```
Decimal('6.127E-13')
```


Alejandro C. Parra Garcia

With normal double precision

```
[133] a = float(-12345678987654321)
      b = float(123)

      solution = a + math.sqrt((a**2)+(b**2))
      f'{solution:.4}'

      '0.0'
```

```
[135] f'{a:.33}'

      '-12345678987654320.0'
```