Homework #1

Numerical Analysis for Materials

POSTECH
POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY
포항공과대학교

학과: 신소재공학과
학번: 20110085
이름: 도경연

MSE POSTECH Materials Science and Engineering

개인 과제물

개인 과제물 (예습)

- Case1_1: 0.00001을 백만 번 더하면서 매 십만 번째마다 결과를 출력하시오.
- Case1_2: 1을 백만 번 더하면서 매 십만 번째마다 결과/100000을 출력하시오.

개인 과제물 (복습)

- Case1_3: 현재 사용하고 있는 컴퓨터가 single precision 에서 구분할 수 있는 수의 정밀도가 어느 정도인지 프로그램을 짜서 확인하고, 이 system 이 몇 개의 bit 로 가수(mantissa)를 표현하는지 판정하시오.

Case 1_1&2

Core Code

```
void Case1(double add_val, long tot_num ,long freq_view, long div_sum)
{
    long count_num; //variable of counting number for "for loop"
    double sum;     //variable of summation with additive value
    double view;    //variable for showing divided summation
    for(count_num = 1, sum = 0;count_num <= tot_num;count_num+=1)
    {
        sum+=add_val;
        if(count_num%freq_view == 0)
        {
            view=sum/div_sum;
            printf("%.17f\n",view);
        }
    }
    printf("\n");
}
```

```
int main()
{
    Case1_1();
    Case1_2();
    return 0;
}
```

```
/*
** add_val: additive value
** tot_num: total number of summation
** freq_view: frequency of showing summation
** div_sum: divider of summation
** Case1(): function of summation and print
*/
```

Result

```
void Case1_1()
{
    Case1(0.00001,1000000,100000,1);
}
```

```
void Case1_2()
{
    Case1(1,1000000,100000,100000);
}
```

```
0.99999999999808376
2.00000000000463500
3.00000000001118620
4.00000000001773690
4.99999999997987920
5.99999999994202150
6.99999999990416380
7.99999999986630610
8.99999999982844830
9.99999999979059060
```

```
1.00000000000000000
2.00000000000000000
3.00000000000000000
4.00000000000000000
5.00000000000000000
6.00000000000000000
7.00000000000000000
8.00000000000000000
9.00000000000000000
10.00000000000000000
```

conclusion

Case1_1:

$$\sum_{i=1}^{100000} 0.00001 \neq 1$$

→ Error in decimal calculation

Case1_2:

$$\left(\sum_{i=1}^{100000} 1\right)/100000 = 1$$

→ No error in integer calculation

Case 1_3

Core Code

```
int num_bit()
{
    float mantissa = 1;
    int count_bit = 0;
    for(;mantissa != 1+mantissa/2;)
    {
        mantissa = 1+mantissa/2;
        count_bit += 1;
    }
    return count_bit-1;
}
```

Code Explanation

# of step	before		# of step	after	
	A	B		A	B
0	1.0	0	**1**	1+0.1	1
1	1.1	1	**2**	1+0.11	2
			...		
22	1.1…1	22	**23**	1+0.1…11	23
23	1.1…11	23	**24**	1+1.0	24

A: mantissa
B: count_bit

Result

```
Decimal = 1.50000000000000000
binary  = 1.1
Decimal = 1.75000000000000000
binary  = 1.11
Decimal = 1.87500000000000000
binary  = 1.111
Decimal = 1.93750000000000000
binary  = 1.1111
Decimal = 1.96875000000000000
binary  = 1.11111
Decimal = 1.98437500000000000
binary  = 1.111111
Decimal = 1.99218750000000000
binary  = 1.1111111
Decimal = 1.99609375000000000
binary  = 1.11111111
Decimal = 1.99804687500000000
binary  = 1.111111111
Decimal = 1.99902343750000000
binary  = 1.1111111111
Decimal = 1.99951171875000000
binary  = 1.11111111111
Decimal = 1.99975585937500000
binary  = 1.111111111111
Decimal = 1.99987792968750000
binary  = 1.1111111111111
Decimal = 1.99993896484375000
binary  = 1.11111111111111
Decimal = 1.99996948242187500
binary  = 1.111111111111111
Decimal = 1.99998474121093750
binary  = 1.1111111111111111
Decimal = 1.99999237060546870
binary  = 1.11111111111111111
Decimal = 1.99999618530273440
binary  = 1.111111111111111111
Decimal = 1.99999809265136720
binary  = 1.1111111111111111111
Decimal = 1.99999904632568360
binary  = 1.11111111111111111111
Decimal = 1.99999952316284180
binary  = 1.111111111111111111111
Decimal = 1.99999976158142090
binary  = 1.1111111111111111111111
Decimal = 1.99999988079071040
binary  = 1.11111111111111111111111
Decimal = 2.00000000000000000
binary  = 10

number of bit for single_mantissa is 23
```

23

conclusion

Case1_3:

- Single precision에서 이 컴퓨터는 $\frac{1}{2^{23}} \approx 1.2 \times 10^{-7}$의 정밀도를 보인다.
- 이 시스템은 single precision에서 mantissa로 23개의 bit를 사용한다.

Thank you

사용 IDE

- Name: Code::Blocks
- Version: Release 16.01 rev 10702
- SDK Version: 1.29.0