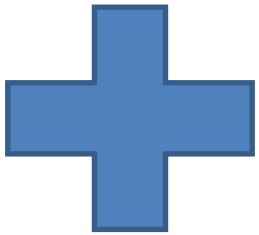
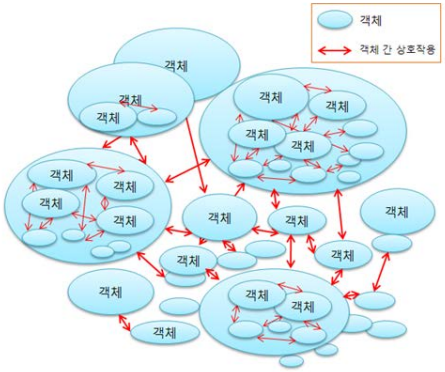
Numerical Method Hw01

20110079 조현렬

- Laguage : C++

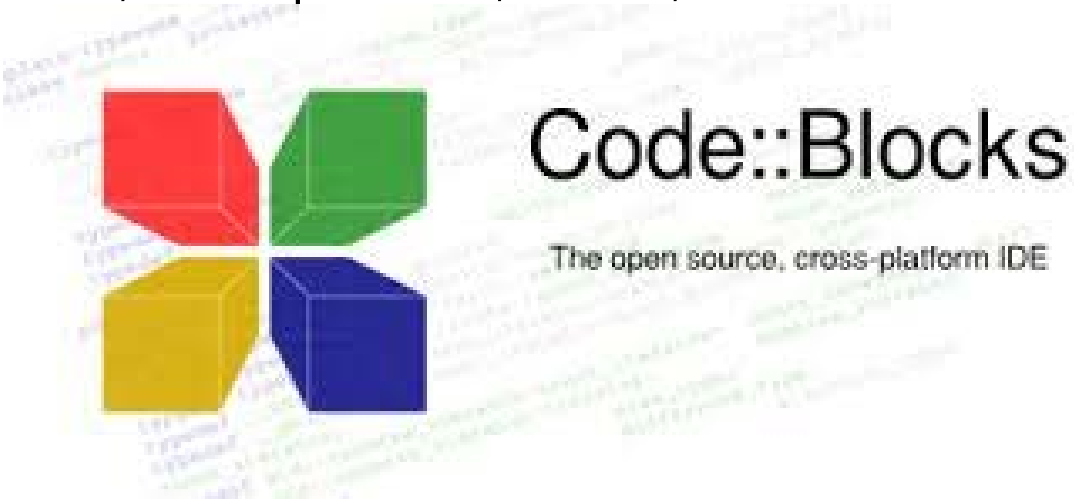


- OOP (Object-Oriented Programming)



- Program : Code blocks

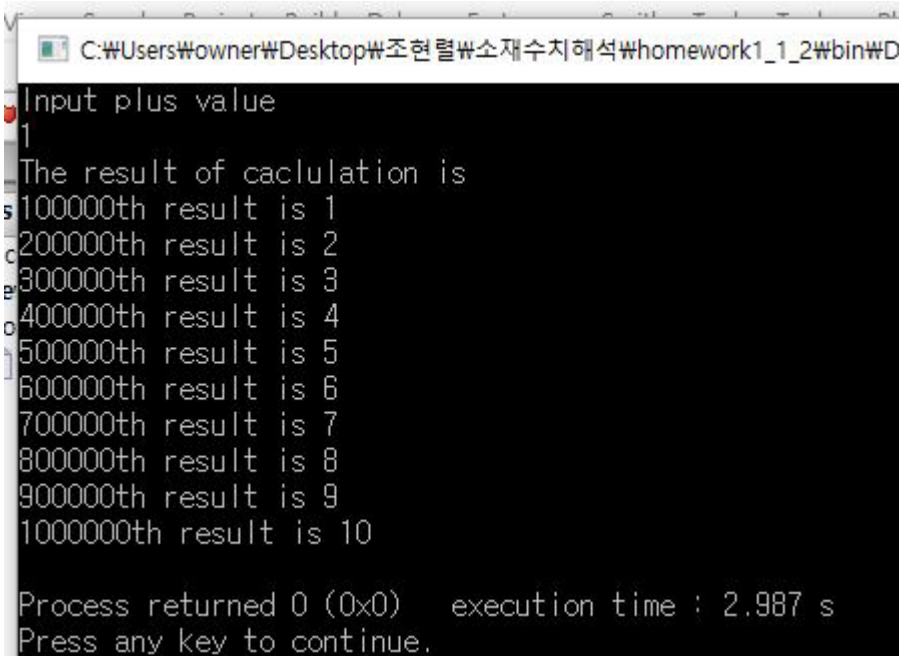
(The open source, cross platform, free C, C++ and Fortran IDE)



Hw1-1

- 1 을 백만 번 더하면서 매 십만 번째마다 결과/100000 을 출력하시오.

```
5   int main()
6   {
7       int i;
8       double sum = 0;
9       double plus_value;
10      double print_out;
11          cout << "Input plus value \n";
12          cin  >> plus_value;
13          cout << "The result of caclulation is \n";
14      for(i=1;i<1000001;i++){
15          sum = sum +plus_value;
16          if(i%100000==0){
17              print_out = sum/100000;
18              cout <<i <<"th result is ";
19              cout << print_out << endl;
20          }
21      }
```



Hw1-1

- 0.00001 을 백만 번 더하면서 매 십만 번째마다 결과를 출력하시오.

```cpp
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int i;
    float f;
    for(i=1;i<1000001;i++){
        f+=0.00001;
        if(i%100000==0){
            cout << i << "th result is ";
            cout << setprecision(30) << f <<endl;
        }
    }
    return 0;
}
```

```
C:₩Users₩owner₩Desktop₩조현렬₩소재수치해석₩homework1₩hw1-
100000th result is 1.000990152359008789062
200000th result is 2.002348184585571289062
300000th result is 3.003706216812133789062
400000th result is 4.005064010620117187
500000th result is 5.006422042846679687
600000th result is 6.007780075073242187
700000th result is 7.009138107299804687
800000th result is 8.009996414184570312
900000th result is 8.963670730590820312
1000000th result is 9.917345046997070312

Process returned 0 (0x0)   execution time : 0.034 s
Press any key to continue.
```

오차율 0.8266% ≒ 1%

Hw1-1

- 0.00001 을 백만 번 더하면서 매 십만 번째마다 결과를 출력하시오.

```
C:₩Users₩owner₩Desktop₩조현렬₩소재수치해석₩homework1₩hw1-2₩main.exe
100000th result is 0.99999999999808375505949697981122881174087524414062 5
200000th result is 2.00000000000463495908320510352496057748794555664062 5
300000th result is 3.00000000001118616310691322723869234323501586914062 5
400000th result is 4.00000000001773692304141150088980793952941894531 25
500000th result is 4.99999999997987920608011336298659443855285644531 25
600000th result is 5.99999999994202148911881522508338093757629394531 25
700000th result is 6.99999999990416377215751708718016743659973144531 25
800000th result is 7.99999999986630605519621894927695393562316894531 25
900000th result is 8.99999999982844833823492081137374043464660644531 25
1000000th result is 9.99999999979059062127362267347052693367004394531 25
```

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int i;
    double f;
    for(i=1;i<1000001;i++){
        f+=0.00001;
        if(i%100000==0){
            cout << i << "th result is ";
            cout << setprecision(100) << f <<endl;
        }
    }
    return 0;
}
```

오차율 0.00001%

결론 : double precision으로 계산하면 생기는 오차는 크게 신경쓰지 않아도 된다

Hw1-2

- 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.

1을 2진수로 표시하면

0 01111111 10000000000000000000000

2로 나누게 되면

0 01111111 01000000000000000000000

Keep going

0 01111111 00000000000000000000000 = 0 (10진수)

매우 작은 숫자를 0으로 계산할 때까지 step수 = mantissa

Profit!!!!!!!

(단 0과 비교하면 exponent value도 줄어들어서 모든 bit가 0이 될때까지 수행함으로 1과 1+delta를 비교)

Hw1-2

- 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.

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int i;
    float a= 1;
    float b= 1;
    for(i=1;i<100001;i++){
     a= a/2;
     b= a+b;
     cout <<i <<"th result is ";
     cout << std::setprecision(1110) << b<< endl;
     if(b==1){
         cout << "Mentissa of my computer is" << i-1<<endl;
         return 0;
     }
     b=1;
    }

     return 0;
}
```

```
C:₩Users₩owner₩Desktop₩조현렬₩소재수치해석₩homev
1th result is 1.5
2th result is 1.25
3th result is 1.125
4th result is 1.0625
5th result is 1.03125
6th result is 1.015625
7th result is 1.0078125
8th result is 1.00390625
9th result is 1.001953125
10th result is 1.0009765625
11th result is 1.00048828125
12th result is 1.000244140625
13th result is 1.0001220703125
14th result is 1.00006103515625
15th result is 1.000030517578125
16th result is 1.0000152587890625
17th result is 1.00000762939453125
18th result is 1.00000381469726562 5
19th result is 1.00000190734863281 25
20th result is 1.00000095367431640625
21th result is 1.00000047683715820312 5
22th result is 1.00000023841857910156 25
23th result is 1.00000011920928955078125
24th result is 1
Mentissa of my computer is23
```

Mantissa of float = 23

Hw1-2

- 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.

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
    int i;
    double a= 1;
    double b= 1;
    for(i=1;i<100001;i++){
     a= a/2;
     b= a+b;
     cout <<i <<"th result is ";
     cout << std::setprecision(1110) << b<
     if(b==1){
         cout << "Mentissa of my computer
         return 0;
     }
     b=1;
    }

     return 0;
```

```
C:₩Users₩owner₩Desktop₩조현렬₩소재수치해석₩homework1₩homework1_1_1₩main....
34th result is 1.00000000005820766091346740722656 25
35th result is 1.00000000002910383045673370361328125
36th result is 1.00000000001455191522836685180664 0625
37th result is 1.00000000000727595761418342590332 03125
38th result is 1.00000000000363797880709171295166 015625
39th result is 1.00000000000181898940354585647583 0078125
40th result is 1.00000000000090949470177292823791 50390625
41th result is 1.00000000000045474735088646411895 751953125
42th result is 1.00000000000022737367544323205947 8759765625
43th result is 1.00000000000011368683772161602973 93798828125
44th result is 1.00000000000005684341886080801486 968994140625
45th result is 1.00000000000002842170943040400743 4844970703125
46th result is 1.00000000000001421085471520200371 74224853515625
47th result is 1.00000000000000710542735760100185 871124267578125
48th result is 1.00000000000000355271367880050092 93556213378906 25
49th result is 1.00000000000000177635683940025046 46778106689453125
50th result is 1.00000000000000088817841970012523 23389053344726562 5
51th result is 1.00000000000000044408920985006261 61694526673632812 5
52th result is 1.00000000000000022204460492503130 80847263336181640625
53th result is 1
Mentissa of my computer is52

Process returned 0 (0x0)   execution time : 0.051 s
Press any key to continue.
```

Mantissa of double = 52