

## Hw2 헤더파일

p.1

Cpp용 time 헤더파일(무작위 용)

X좌표와 Y좌표를 저장할 cold 구조체 생성

좌표들

계산용 클래스 생성

최소 거리, 최대 거리 저장 함수

숫자 입력 받는 함수(문자로 입력 받고 실수 반환)  
자릿수 계산하는 함수(input에 내장시켜 현재는 안씀)

제곱근 계산기

```
1  #include <iostream>
2  #include <ctime>
3
4  using namespace std;
5
6  struct cold
7  {
8      int X;
9      int Y;
10 };
11
12 class pross{
13     private:
14
15         float mid,mad;
16
17     public:
18         float input();
19         int muld(int a);
20         double sqrt(int a);
21
22     };
```

Hw2  
구동부  
sqrt  
p.2

```
1  #include "../include/cpp_test/hw_2.hpp"
2
3
4  ✓ double pross::sqrt (int a){
5      double t=0;
6      while (a>(t*t)){t++;}
7      t--;
8      while (a>(t*t)){t+=0.0001;}
9      t-=0.0001;
10     return t;
11 }
12
```

제공된 공식 a로 입력 받음

입력 받은 수가  $t^2$ 보다 작아질 때까지 t 1씩 증가  
 $a < t^2$ 이면 t-1을 함(a보다 커지기 직전으로 돌림)

더 정밀하게 측정하기 위해 0.0001씩 증가

위와 동일

T 반환

```

14  float pross::input(){
15      char s[1000]={0};
16      int i=0,sig=0,k=0;
17      cin>>s;
18      if(s[0]=='-'){sig=1;i=1;}
19      if(s[0]==0){return 0.1;}
20      while (1)
21      {
22          if(s[i]==0){
23              int mult=1,a=0;
24              while (k+sig<i){
25                  a+=(s[i-k-1]-'0')*mult;
26                  k++;
27                  mult*=10;
28              }
29              if(sig==1){a=-a;}
30              return a;
31          }
32          else if('0'>s[i]||s[i]>'9'){return 0.1;}
33          i++;
34      }
35  }

```

입력 처리  
글자 받을 배열 생성 및 초기화

배열에 입력 담기

만약 첫번째 문자가 -면 숫자를 1부터 읽으면서(i)

음수라는 신호 on(sig)

아무것도 없으면 0.1 보내기

(입력을 int로만 입력 받기에 처리 과정에서 실수 들어오면 재입력하게 됨) (근데 cin은 띄어쓰기나 엔터를 입력으로 처리 안하여 솔직히 필요 없음)

'w0'이 나올 때까지 반복하여 숫자 길이 측정(i가 측정함)

A는 배열의 i번째부터 1칸씩 배열의 수를 읽으며 해당 자리 수 만큼 10을 곱하면서 더해짐, 이때 sig+k가 총 자릿수(i)만큼 커질때 까지 반복

만약 음수면 음수로 반전하고 a를 반환

만일 중간에 숫자 이외의 것이 나오면 0.1로 반환

Hw2  
구동부  
input  
p.3

```

38  ✓ int main(){
39
40      srand((unsigned int)time(NULL));
41
42      pross point;
43      int i,mi,ma,sh,sh1,lo,lo1;
44      float p=0.1,min=0.1,max=0.1;
45
46      while(p<1||(p-int(p)!=0)){//입력들(개수, 최대, 최소)
47          cout<<"Please define the number of points(natural number): ";
48          p=point.input();
49      }
50      i=p;
51      while(min-(int)min!=0){
52          cout<<"Please define minimum of coor. value(int): ";
53          min=point.input();
54      }
55      mi=min;
56      while(max<min||(max-(int)max!=0)){
57          cout<<"Please define maximum of coor. value(bigger int than min): ";
58          max=point.input();
59      }
60      ma=max;
61
62      cout<<endl<<endl<<"Generate Random point"<<endl;
63
64      float mid,mad,len;
65      cold *list=new cold[i];

```

시간으로 무작위 시드설정

클래스 생성

포인트 개수, 최소 좌표, 최대 좌표 초기화

자연수가 아니면 계속 입력 받게 함

Input을 통해 처리

P를 i로 바꿔 int형으로 바꿈

최소 좌표는 정수일 때까지 계속 입력 받음

최대는 min보다 크고 정수일 때까지 입력 받음

I의 개수만큼 cold 구조체로 된 동적 배열 만듦

Hw2  
구동부  
main  
p.4

Hw2  
구동부  
main  
p.5

```
7  for(int k=0;i>k;k++){
8      int d=0;
9      while(d==0){//겹치면 다시 돌리고 더 이상 새로운 좌표 없으면 겹치기
10         list[k].X=(rand()%(ma-mi+1))+mi; 구조체 K번째 배열의 X변수에 범위 안에서 랜덤으로 값 생성
11         list[k].Y=(rand()%(ma-mi+1))+mi; Y도 마찬가지로
12         if(k<(ma-mi+1)*(ma-mi+1)){ 만일 범위 안에 가능한 좌표의 값 이상의 좌표 생성이 안됐다면
13             for(int j=0;j<k;j++){ 좌표 리스트 검사(겹치는지 확인)
14                 if((list[k].X==list[j].X)&&(list[k].Y==list[j].Y)){
15                     d=0; 만일 겹치는 것이 있으면 while문으로 다시 생성
16                     break;
17                 }
18                 else{d=1;} 아니면 d=1로 바꿔 통과하도록 만들
19             }
20             if(k==0){d=1;} 만일 첫번째 좌표라면 그냥 통과
21         }
22         else{d=1;} 범위 내에 가능한 모든 좌표<=생성 수 일시 그냥 통과
23     }
24     cout<<"Point " <<k+1<<" . nX="<<list[k].X<<" , nY="<<list[k].Y<<endl; 좌표 출력
25 }
26 for(int k=0;i>k;k++){
27     for(int j=1+k; i>j;j++){ 모든 좌표간 거리 구함(sqrt로 해결)
28         len=point.sqrt(((list[k].X-list[j].X)*(list[k].X-list[j].X))+((list[k].Y-list[j].Y)*(list[k].Y-list[j].Y)));
29         if(j==1){mid=210000000;mad=0;}//최소에 매우 큰수, 최대에 작은 수
30         if (len<0){len=0;}
31         if(mad<len){mad=len;lo=k;lo1=j;} 최대, 최소 갱신할 때마다 해당 list 번호로 좌표저장 및 길이 저장
32         if(mid>len){mid=len;sh=k;sh1=j;}
33     }
34 }
```

과제에 나온대로 출력한 후 list 동적 할당 반환

```
95 //과제 나온대로 출력
96     cout<<endl<<"-----Result-----"<<endl;
97     cout<<"MinDist="<<mid<<endl;
98     cout<<"Pair of Min Coord.<x,y>: P1"<<list[sh].X<<" "<<list[sh].Y<<"> & P2"<<list[sh1].X<<" "<<list[sh1].Y<<">"<<endl<<endl;
99     cout<<"MinDist="<<mad<<endl;
00     cout<<"Pair of Max Coord.<x,y>: P1"<<list[lo].X<<" "<<list[lo].Y<<"> & P2"<<list[lo1].X<<" "<<list[lo1].Y<<">"<<endl<<endl<<endl;
01     cout<<"*****Completed*****"<<endl<<endl<<endl;
02     cout<<"Press <RETURN> to close this window...";
03
04
05 }
06 delete []list;
```