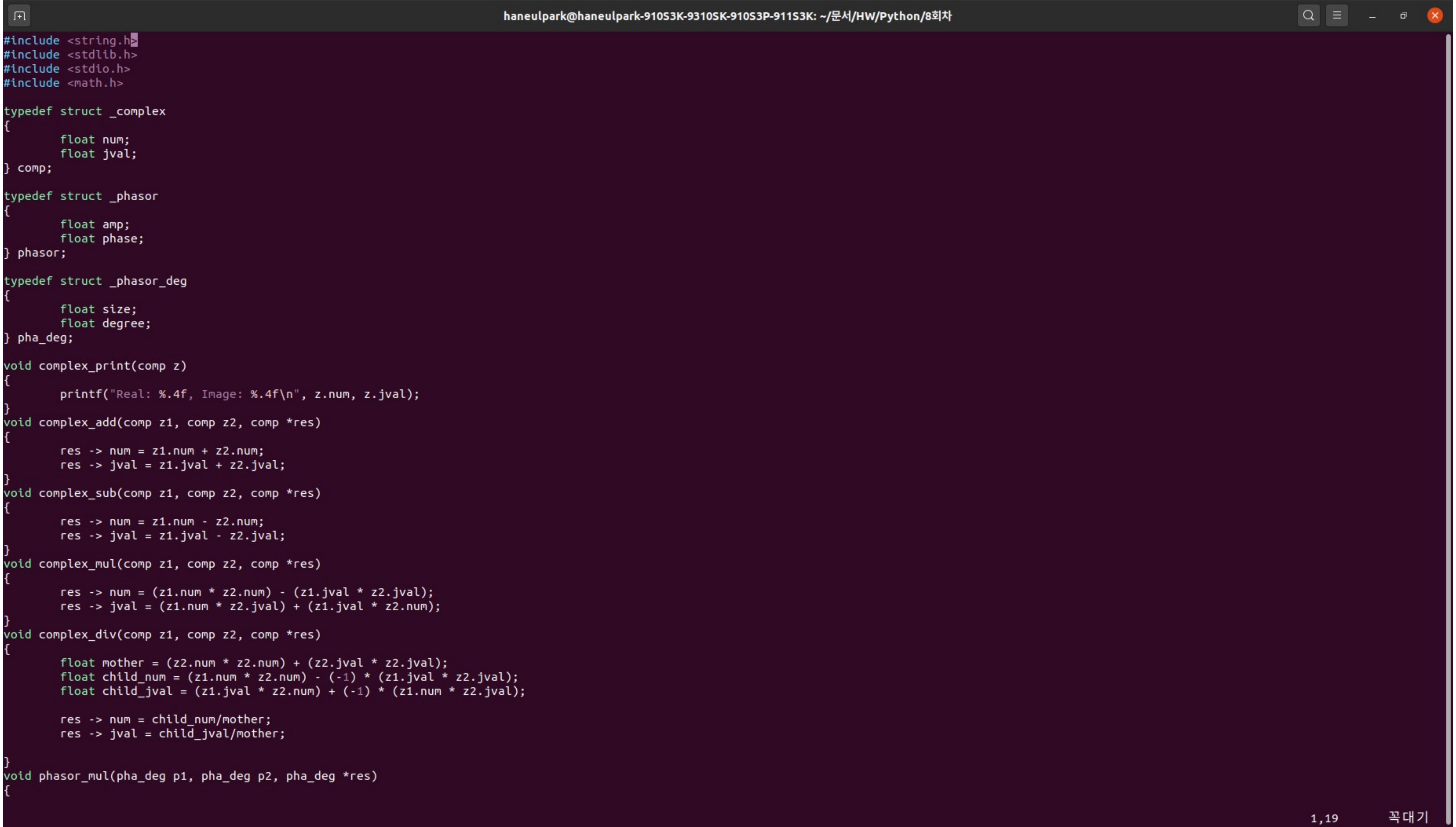
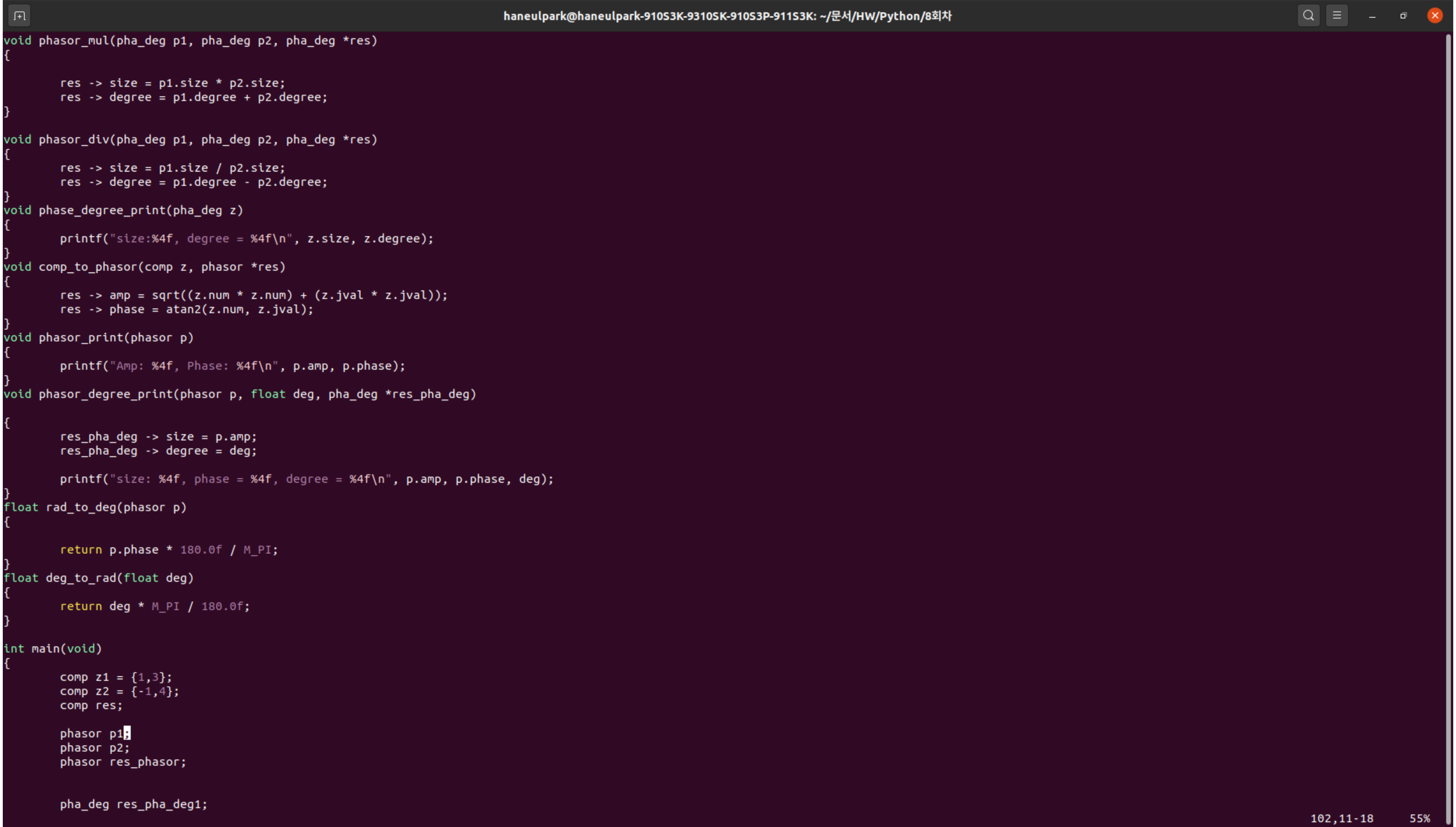
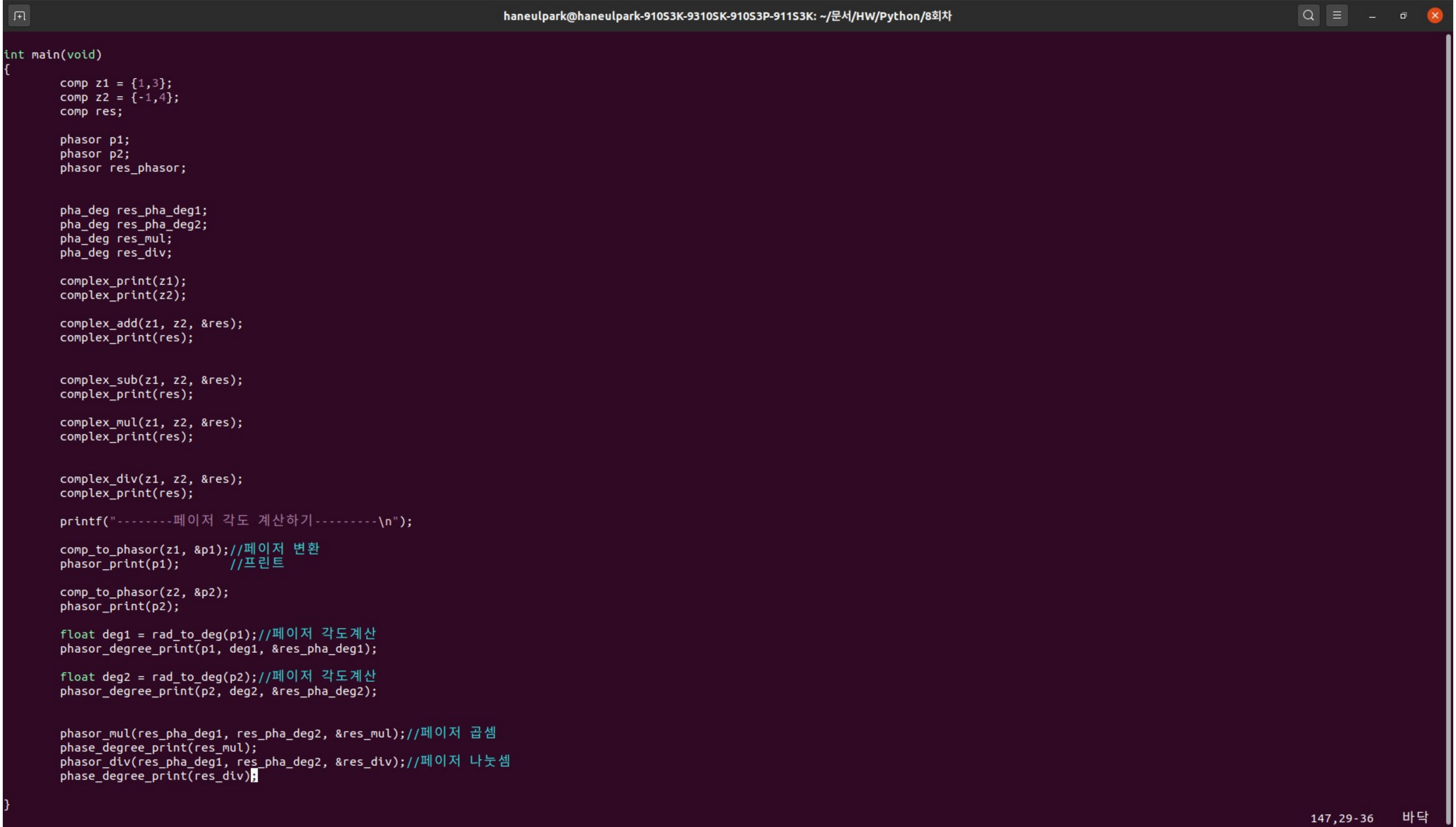

haneulpark@haneulpark-910S3K-9310SK-910S3P-911S3K: ~/문서/HW/Python/8회차
1,19
꼭대기

```
#include <string.h>
#include <stdlib.h>
#include <stdio.h>
#include <math.h>

typedef struct _complex
{
        float num;
        float jval;
} comp;

typedef struct _phasor
{
        float amp;
        float phase;
} phasor;

typedef struct _phasor_deg
{
        float size;
        float degree;
} pha_deg;

void complex_print(comp z)
{
        printf("Real: %.4f, Image: %.4f\n", z.num, z.jval);
}
void complex_add(comp z1, comp z2, comp *res)
{
        res -> num = z1.num + z2.num;
        res -> jval = z1.jval + z2.jval;
}
void complex_sub(comp z1, comp z2, comp *res)
{
        res -> num = z1.num - z2.num;
        res -> jval = z1.jval - z2.jval;
}
void complex_mul(comp z1, comp z2, comp *res)
{
        res -> num = (z1.num * z2.num) - (z1.jval * z2.jval);
        res -> jval = (z1.num * z2.jval) + (z1.jval * z2.num);
}
void complex_div(comp z1, comp z2, comp *res)
{
        float mother = (z2.num * z2.num) + (z2.jval * z2.jval);
        float child_num = (z1.num * z2.num) - (-1) * (z1.jval * z2.jval);
        float child_jval = (z1.jval * z2.num) + (-1) * (z1.num * z2.jval);

        res -> num = child_num/mother;
        res -> jval = child_jval/mother;

}
void phasor_mul(pha_deg p1, pha_deg p2, pha_deg *res)
{
```


haneulpark@haneulpark-910S3K-9310SK-910S3P-911S3K: ~/문서/HW/Python/8회차

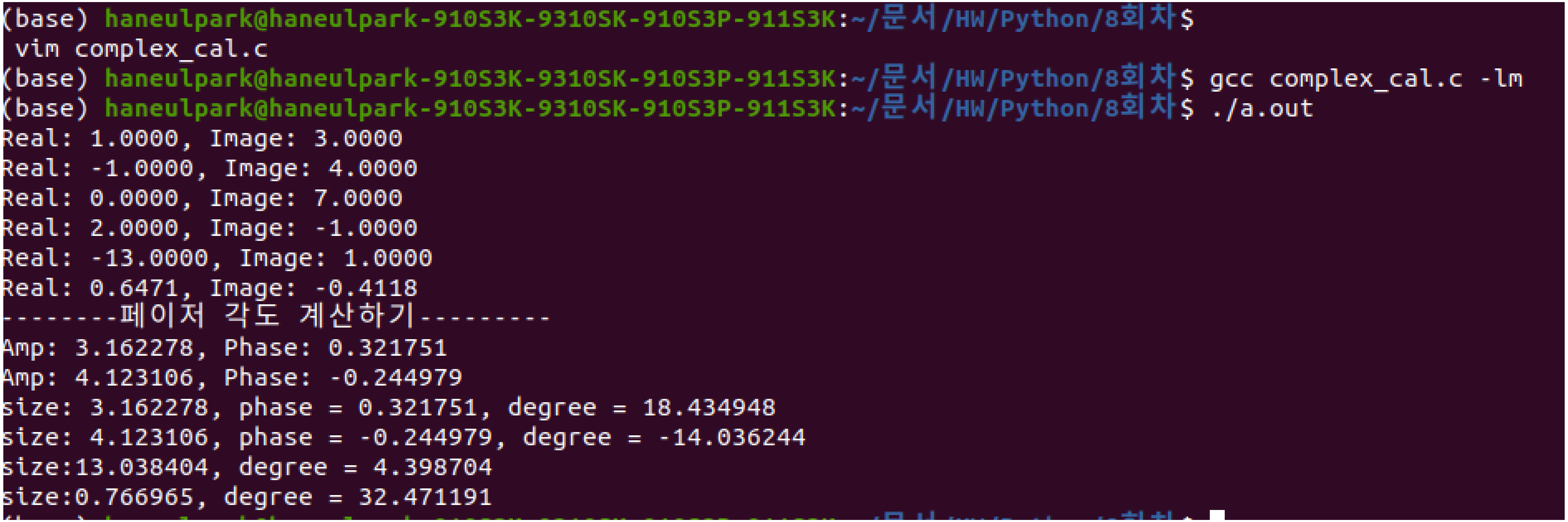
```
void phasor_mul(pha_deg p1, pha_deg p2, pha_deg *res)
{

        res -> size = p1.size * p2.size;
        res -> degree = p1.degree + p2.degree;
}

void phasor_div(pha_deg p1, pha_deg p2, pha_deg *res)
{
        res -> size = p1.size / p2.size;
        res -> degree = p1.degree - p2.degree;
}
void phase_degree_print(pha_deg z)
{
        printf("size:%4f, degree = %4f\n", z.size, z.degree);
}
void comp_to_phasor(comp z, phasor *res)
{
        res -> amp = sqrt((z.num * z.num) + (z.jval * z.jval));
        res -> phase = atan2(z.num, z.jval);
}
void phasor_print(phasor p)
{
        printf("Amp: %4f, Phase: %4f\n", p.amp, p.phase);
}
void phasor_degree_print(phasor p, float deg, pha_deg *res_pha_deg)

{
        res_pha_deg -> size = p.amp;
        res_pha_deg -> degree = deg;

        printf("size: %4f, phase = %4f, degree = %4f\n", p.amp, p.phase, deg);
}
float rad_to_deg(phasor p)
{

        return p.phase * 180.0f / M_PI;
}
float deg_to_rad(float deg)
{
        return deg * M_PI / 180.0f;
}

int main(void)
{
        comp z1 = {1,3};
        comp z2 = {-1,4};
        comp res;

        phasor p1;
        phasor p2;
        phasor res_phasor;


        pha_deg res_pha_deg1;
```



haneulpark@haneulpark-910S3K-9310SK-910S3P-911S3K: ~/문서/HW/Python/8회차

```
int main(void)
{
        comp z1 = {1,3};
        comp z2 = {-1,4};
        comp res;

        phasor p1;
        phasor p2;
        phasor res_phasor;


        pha_deg res_pha_deg1;
        pha_deg res_pha_deg2;
        pha_deg res_mul;
        pha_deg res_div;

        complex_print(z1);
        complex_print(z2);

        complex_add(z1, z2, &res);
        complex_print(res);


        complex_sub(z1, z2, &res);
        complex_print(res);

        complex_mul(z1, z2, &res);
        complex_print(res);


        complex_div(z1, z2, &res);
        complex_print(res);

        printf("--------페이저 각도 계산하기---------\n");

        comp_to_phasor(z1, &p1);//페이저 변환
        phasor_print(p1);       //프린트

        comp_to_phasor(z2, &p2);
        phasor_print(p2);

        float deg1 = rad_to_deg(p1);//페이저 각도계산
        phasor_degree_print(p1, deg1, &res_pha_deg1);

        float deg2 = rad_to_deg(p2);//페이저 각도계산
        phasor_degree_print(p2, deg2, &res_pha_deg2);


        phasor_mul(res_pha_deg1, res_pha_deg2, &res_mul);//페이저 곱셈
        phase_degree_print(res_mul);
        phasor_div(res_pha_deg1, res_pha_deg2, &res_div);//페이저 나눗셈
        phase_degree_print(res_div);

}
```

result



```
(base) haneulpark@haneulpark-910S3K-9310SK-910S3P-911S3K:~/문서/HW/Python/8회차$
 vim complex_cal.c
(base) haneulpark@haneulpark-910S3K-9310SK-910S3P-911S3K:~/문서/HW/Python/8회차$ gcc complex_cal.c -lm
(base) haneulpark@haneulpark-910S3K-9310SK-910S3P-911S3K:~/문서/HW/Python/8회차$ ./a.out
Real: 1.0000, Image: 3.0000
Real: -1.0000, Image: 4.0000
Real: 0.0000, Image: 7.0000
Real: 2.0000, Image: -1.0000
Real: -13.0000, Image: 1.0000
Real: 0.6471, Image: -0.4118
--------페이저 각도 계산하기---------
Amp: 3.162278, Phase: 0.321751
Amp: 4.123106, Phase: -0.244979
size: 3.162278, phase = 0.321751, degree = 18.434948
size: 4.123106, phase = -0.244979, degree = -14.036244
size:13.038404, degree = 4.398704
size:0.766965, degree = 32.471191
```