

# myNP.h

```
#include "../include/myNP.h"
```

자신의 폴더에 위치에 있는 myNP.h라는 헤더파일 가져오는 방법

```
#define PI 3.14159265358979323846264338327950288419716939937510582
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
```

PI 값 정의 실시

기본 library를 include 시키기

## 중간고사 함수

- [laplacianGaussian\(\)](#)

### laplacianGaussian()

laplacianGaussian의 값을 구한다.

x, y를 편미분을 하고 각각 편미분 한 것을 더한다.

이때 two-point forward의 방식을 이용해서 해를 구한다.

$$\frac{\delta f^2(x, y)}{\delta x^2} = \frac{f(x_{i-1}, y_i) - 2f(x_i, y_i) + f(x_{i+1}, y_i)}{h_x^2}$$
$$\frac{\delta f^2(x, y)}{\delta y^2} = \frac{f(x_i, y_{i-1}) - 2f(x_i, y_i) + f(x_i, y_{i+1})}{h_y^2}$$
$$laplacianGaussian = \frac{\delta f^2(x, y)}{\delta x^2} + \frac{\delta f^2(x, y)}{\delta y^2}$$

```
double laplacianGaussian(double func(double x, double y), double _x, double _y);
```

#### Parameters

func(double x, double y): 구하고 싶은 함수

\_x: x 변수

\_y: y 변수

#### Example code

```
#include "../include/myNP.h"
double q4_func(double _x, double _y)
{
```

```

double out = 0.;
double x = _x;
double y = _y;
double sigma = 3.0;
out = (1 / (2 * PI * sigma * sigma) * exp(-(x * x + y * y) / (2 * sigma *
sigma)));
return out;
}
int main(){
double x = 1.0;
double y = 0.0;
double LoG = 0;
LoG = laplacianGaussian(q4_func, x, y);
printf("\n LoG at [1][0] = %.4f \t\n\n", LoG);
return 0;
}

```

### output

```
LoG at [1][0] = -0.0034
```

### Warning

- x는 double의 형식이어야 한다.