TI DSP, MCU, Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee (이상훈)
gcccompil3r@gmail.com
학생 – 김형주
mihaelkel@naver.com

matrix.c

```c
#include <stdio.h>
#include <math.h>
void print_mat(double (*mat)[3]);
void add_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3]);
void sub_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3]);
void mul_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3]);
double det_mat(double (*mat)[3]);
void adjoint_mat(double (*mat)[3],double (*adjMat)[3]);
void reverse_mat(double (*mat)[3], double (*res)[3]);
int main(void)
{
    double mat1[3][3] = {{1,2,3},{2,5,3},{1,0,8}};
    double mat2[3][3] = {{2,0,4},{0,3,9},{0,0,1}};
    double res[3][3] = {{1,2,4},{16,8,4},{2,2,2}};
    reverse_mat(mat1, res);
    print_mat(res);
}
void print_mat(double (*mat)[3])
{
    int i, j;
    for(i = 0; i < 3; i++)
    {
        for(j=0;j<3;j++)
            printf("%lf\t",mat[i][j]);
        printf("\n");
    }
}
void add_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3])
{
    int i,j;
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            res[i][j] = mat1[i][j] + mat2[i][j];
}
void sub_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3])
{
    int i,j;
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            res[i][j] = mat1[i][j] - mat2[i][j];
}
void mul_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3])
{
    int i,j,k;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            res[i][j] = 0;
            for(k=0;k<3;k++)
                res[i][j] += mat1[i][k]*mat2[k][j];
        }
    }
```

```
}
double det_mat(double (*mat)[3])
{
    double res = 0;
    int i;
    for(i=0;i<3;i++)
        res += (mat[1][(i+1)%3]*mat[2][(i+2)%3] - mat[1][(i+2)%3]*mat[2][(i+1)%3])*mat[0][i];
    return res;
}
void adjoint_mat(double (*mat)[3],double (*adjMat)[3])
{
    int i, j;

    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            adjMat[j][i]                =                (mat[(i+1)%3][(j+1)%3]*mat[(i+2)%3][(j+2)%3]                -
mat[(i+2)%3][(j+1)%3]*mat[(i+1)%3][(j+2)%3]);


}
void reverse_mat(double (*mat)[3], double (*res)[3])
{
    double det;
    double adjMat[3][3];
    int i,j;
    det = det_mat(mat);
    adjoint_mat(mat,adjMat);
    for(i=0;i<3;i++)
        for(j=0;j<3;j++)
            res[i][j] = adjMat[i][j] / det;

}
```