

TI DSP, MCU, Xilinx Zynq FPGA 프로그래밍 전문가 과정

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

matrix.c

```
1  #include <stdio.h>
2  #include <math.h>
3  void print_mat(double (*mat)[3]);
4  void add_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3]);
5  void sub_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3]);
6  void mul_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3]);
7  double det_mat(double (*mat)[3]);
8  void adjoint_mat(double (*mat)[3], double (*adjMat)[3]);
9  void reverse_mat(double (*mat)[3], double (*res)[3]);
10 int main(void)
11 {
12     double mat1[3][3] = {{1,2,3},{2,5,3},{1,0,8}};
13     double mat2[3][3] = {{2,0,4},{0,3,9},{0,0,1}};
14     double res[3][3] = {{1,2,4},{16,8,4},{2,2,2}};
15     reverse_mat(mat1, res);
16     print_mat(res);
17 }
18 void print_mat(double (*mat)[3])
19 {
20     int i, j;
21     for(i = 0; i < 3; i++)
22     {
23         for(j=0;j<3;j++)
24             printf("%lf\t",mat[i][j]);
25         printf("\n");
26     }
27 }
28 void add_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3])
29 {
30     int i,j;
31     for(i=0;i<3;i++)
32         for(j=0;j<3;j++)
33             res[i][j] = mat1[i][j] + mat2[i][j];
34 }
35 void sub_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3])
36 {
37     int i,j;
38     for(i=0;i<3;i++)
39         for(j=0;j<3;j++)
40             res[i][j] = mat1[i][j] - mat2[i][j];
41 }
42 void mul_mat(double (*mat1)[3], double (*mat2)[3], double (*res)[3])
43 {
44     int i,j,k;
45     for(i=0;i<3;i++)
46     {
47         for(j=0;j<3;j++)
48         {
49             res[i][j] = 0;
50             for(k=0;k<3;k++)
51                 res[i][j] += mat1[i][k]*mat2[k][j];
52         }
53     }
```

```

54
55
56
57
58 }
59 double det_mat(double (*mat)[3])
60 {
61     double res = 0;
62     int i;
63     for(i=0;i<3;i++)
64         res += (mat[1][(i+1)%3]*mat[2][(i+2)%3] - mat[1][(i+2)%3]*mat[2][(i+1)%3])*mat[0][i];
65     return res;
66 }
67 void adjoint_mat(double (*mat)[3],double (*adjMat)[3])
68 {
69     int i, j;
70
71     for(i=0;i<3;i++)
72         for(j=0;j<3;j++)
73             adjMat[j][i] = (mat[(i+1)%3][(j+1)%3]*mat[(i+2)%3][(j+2)%3] -
74 mat[(i+2)%3][(j+1)%3]*mat[(i+1)%3][(j+2)%3]);
75
76 }
77 void reverse_mat(double (*mat)[3], double (*res)[3])
78 {
79     double det;
80     double adjMat[3][3];
81     int i,j;
82     det = det_mat(mat);
83     adjoint_mat(mat,adjMat);
84     for(i=0;i<3;i++)
85         for(j=0;j<3;j++)
86             res[i][j] = adjMat[i][j] / det;
87
88 }
89
90
91
92

```