

問題 1

次のように宣言された二次元配列に格納された整数値を実行例のように二通りで表示するプログラムを作成せよ。ただし、表示には繰り返し処理（for 文）を用いよ。

```
int a[4][4] = {  
    { 1,  2,  3,  4 },  
    { 5,  6,  7,  8 },  
    { 9, 10, 11, 12 },  
    { 13, 14, 15, 16 }  
};
```

出力例：

```
$ ./transpose  
行列 A  
  1  2  3  4  
  5  6  7  8  
  9 10 11 12  
13 14 15 16  
行列 A の 転 置  
  1  5  9 13  
  2  6 10 14  
  3  7 11 15  
  4  8 12 16  
$ █
```

問題 2

次の回転行列を用いると、座標 (x, y) を原点を中心にして反時計回りに θ 回転（ラジアン）

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

ン）させた座標 (x', y') を計算することができる。

この行列を用いてキーボードからもとの座標を入力し、 θ 回転させた座標を出力するプログラムを作成した。下線部を修正して正しく動作するプログラムを作成せよ。ただし、数学関数 \sin, \cos はプログラム中で

```
y = sin(theta); y = cos(theta);
```

のようにして使うことができる（ θ はラジアン）。また、行列ベクトル積には繰り返し処理（for 文）を用いよ。

```
#include <stdio.h>
#include<math.h>
/*_____の部分を埋めてプログラムを完成させよ*/
main()
{
    int i, j;
    float _____; /*回転行列 a の宣言*/
    float theta; /*回転（ラジアン）*/
    float x[2]; /*もとの座標*/
    float y[2]; /*回転後の座標*/

    /*シータの入力*/
    printf(“回転（ラジアン:”);
    scanf(“%f”, &theta);

    /*回転行列 a を作成*/
    a[0][0]=_____ ; a[0][1]=_____ ;
    a[1][0]=_____ ; a[1][1]=_____ ;

    /*もとの座標の入力*/
    printf(“もとの座標:”);
    scanf(“%f %f “, &x[0], &x[1]);
    /*行列・ベクトル積*/
    _____
    _____
    _____
    _____

    /*変換後の座標を表示*/
    printf(“変換後の座標 (%f, %f)\n”, y[0], y[1]);
}
```

出力例：

```
$ ./rotate
回転（ラジアン）：1.570796
もとの座標：1.0 0.0
変換後の座標（0.000000, 1.000000）
$
```

問題 3

プログラム中で与えられている 4x4 の int 型の二次元配列 `square` に格納されている数値からなる表が魔方陣になっているかどうかのプログラムを作成せよ。作成するプログラムは 4x4 の魔方陣をチェックするプログラムで良いが、繰り返し処理（for 文など）を適切に利用して 3x3, 5x5, 6x6, 9x9, ... にも簡単に拡張できるようにせよ。

魔方陣とは、正方形の方陣に数字を配置したときに、以下の条件を満足するものを言う。

条件 1 : 縦・横・斜めいずれの列でも数字の合計が同じ

条件 2 : 1 から方陣のマスの数までの数字を過不足なく使う

4x4 の魔方陣の例

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

上記の例の場合、縦の列の合計は(34, 34, 34, 34), 横の列の合計も (34, 34, 34, 34), 斜めの列の合計も (34, 34) となりすべて等しいため条件 1 を満足する。また、1 から 16 までの数字をすべて一つずつ使った配置となっているため条件 2 を満たす。