

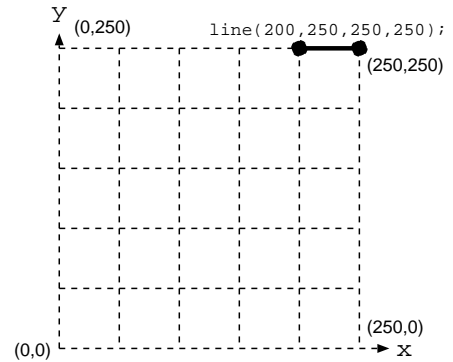
以下の設問から 4 つを選んで回答せよ。

$(x_0, y_0) - (x_1, y_1)$ の 2 点間を直線で結ぶ `line(x0,y0,x1,y1)` 関数があるものとする。
一連の出題で描く画面は、左下を原点とする。

1 基礎問題

```
int table[ 4 ] = { 100 , 150 , 200 , 50 } ;
```

```
void main() {
    int i , j , x = 0 , y = 0 ;
    int* px = &table[3] ; /* 注意 */
    for( i = 0 ; i < 4 ; i++ ) {
        if ( i == 2 ) {
            int y = 100 ;
            line( x , y , x + 100 , y ) ;
        } else {
            for( j = 0 ; j < 2 ; j++ ) {
                if ( i + j >= 4 )
                    break ;
                line( *px , y , *px , y + 50 ) ;
                y += 50 ;
            }
            px-- ; /* 注意 */
        }
    }
}
```



2 説明問題 (25)

以下の説明問題から 2 つを選んで答えよ。

1. 2次元座標の回転行列について説明せよ。
2. 2消点法について図などを交えて説明せよ。
3. コンパイラが行ってくれる最適化について例をあげて説明せよ。

3 配列渡しとファイル (25)

```

void mat_2x2_2( double a[2] , _____(A)3 _____(B)3 ) {
    int i , j ;
    for( i = 0 ; i < 2 ; i++ ) {
        _____(C)3
        for( j = 0 ; j < 2 ; j++ )
            _____(D)5+
    }
}

void main() {
    FILE* fp ;    double xy[ 2 ] , ans[ 2 ] ;

    if ( (fp = _____(E)4 ) != NULL ) {
        while( fscanf( fp , "%lf %lf" , &xy[0] , &xy[1] ) == 2 ) {
            mat_2x2_2( _____(F)3 , rot , xy ) ;
            printf( "%lf,%lf\n" , ans[0] , ans[1] ) ;
        }
    } _____(G)4
}

```

```

#define RAD (3.141592/180.0)
double rot[2][2] = {
    { cos(30*RAD), -sin(30*RAD) } ,
    { sin(30*RAD),  cos(30*RAD) } ,
}

```

(設問)
 | ファイル file.txt に 1 行毎に
 | X 座標,Y 座標で複数行記録され
 | ている。この座標に、回転行列
 | をかけた結果を出力する
 | プログラムを作成した。
 | 下線部 (A)-(G) にふさわしい
 | 処理を記述せよ。

4 遠近法 (25)

画面の中央から奥に進む先に、正方形が並ぶ絵を描きたい。右図を参考に、ふさわしい処理を埋めよ。

```

#define XWIDTH  640  /* 画面幅 */
#define YHEIGHT 480  /* 画面高さ */
#define VP      500.0 /* 視点位置 */
#define SQSIZE  500.0 /* 正方形の 1/2 */

double perspect_px( _____(A)3 _____(B)7 )
{
    double h = _____(C)7
    return _____(D)4
}

void main() {
    double z ;
    for( z = 1000 ; z <= 1200 ; z += 200 ) {
        double px = perspect_px( z , SQSIZE , VP ) ;

        line( XWIDTH/2 - px , YHEIGHT/2 - px , XWIDTH/2 + px , YHEIGHT/2 - px ) ;
        line( XWIDTH/2 - px , YHEIGHT/2 + px , XWIDTH/2 + px , YHEIGHT/2 + px ) ;

        line( _____(E)4 ) ;
        line( _____(E)4 ) ;
    }
}

```

