```
/**
 * 演習問題8: サーボモータと測距センサ使用．距離に応じてサーボモーターの回転角を変える
 * 距離センサ: 10-80cmまで検出, サーボモータは距離[cm]の2倍の角度で回転
 */

#include <Servo.h>  // サーボモータ用ライブラリ

// defineはコンパイル時にマクロ変換される
#define SH_2Y0A21 0  // アナログ距離センサ接続ピン
#define SERVO 9      // サーボモータ接続ピン
Servo servo1;        // 操作するサーボモータ

/**
 * 関数名: setup
 * 引数: なし
 * 処理: 各初期設定のため初回のみ実行
 * 返り値: なし
 */
void setup () {
    // put your setup code here, to run once:

    Serial.begin ( 9600 );    // シリアル通信の初期化
    servo1.attach ( SERVO );  // サーボモータを接続したピンを使用
}

int count = 5;  // 距離センサ測定回数

/**
 * 関数名: loop
 * 引数: なし
 * 処理: 無限ループ．サーボモーターの回転角を距離によって変化させる
 * 返り値: なし
 */
void loop () {
    // put your main code here, to run repeatedly:

    float ain = 0.0;                          // 測定距離平均値

    // 平均を計算するため総和する
    for ( int i = 0; i < count; i++ ) {
        ain += analogRead ( SH_2Y0A21 );
    }
    ain = ain / count;                        // A/D変換値の平均
    float dcm = ( 6787 / ( ain - 3 ) ) - 4;  // A/D変換値平均から距離に直す[cm]

    // 距離の表示
```

```
    Serial.print ( "d: " );
    Serial.print ( dcm );
    Serial.print ( " cm" );

    // 10-80cm以内に障害物があるか
    if ( 10 <= dcm && dcm < 80 ) {

        // 回転角の表示
        Serial.print ( " , rot: " );
        Serial.print ( 2 * dcm );
        Serial.println ( " deg" );

        servo1.write ( 2 * dcm );             // サーボモーター2倍の距離
[cm]度に回転[deg]
    }

    delay ( 2000 );                           // 遅延[ms]
}
```