

課題 3-1～3-5 のプログラム例

```

#include <stdio.h>

/* グローバル変数として宣言する */
int test[5] = {80, 60, 22, 50, 75};

/* 関数のプロトタイプ宣言 */
void show();
void show_range(int s, int e);
void swap_array(int i, int j);
int min_array();
int min_range(int s, int e);
void sort_array();
void sort_array2();

/* 配列 test の要素を出力する */
void show()
{
    int i;
    for(i=0; i<5; i++) {
        printf("test[%d]: %d\n", i, test[i]);
    }
}

/* 配列 test の要素を部分的に出力する */
void show_range(int s, int e)
{
    int i;
    /* s～e 番目までの要素に出力を繰り返す */
    for(i=s; i<=e; i++) {
        printf("test[%d]: %d\n", i, test[i]);
    }
}

/* 要素の値を入れ替える */
void swap_array(int i, int j)
{
    int tmp;
    /* 代入を繰り返して入れ替える */
    tmp = test[i];
    test[i] = test[j];
    test[j] = tmp;
}

/* 配列の中から最小値を持つ要素の場所を求める */
int min_array()
{
    int i, r;
    /* 最初の最小値を 0 番目とする */
    r = 0;
    /* 0～4 番目まで最小値との比較を繰り返す */
    for(i=0; i<5; i++) {
        /* i 番目の要素が最小値より小さい場合、
           最小値を更新する */
        if(test[r] > test[i]) {
            r = i;
        }
    }
    /* 繰り返し終了時の r が最小値の場所となる */
    return r;
}

/* 指定した範囲内で最小値の場所を求める */
int min_range(int s, int e)
{
    int i, r;
    /* 最初の最小値を s 番目の要素とする */
    r = s;
    /* s～e 番目まで最小値との比較を繰り返す */
    for(i=s; i<=e; i++) {
        /* i 番目の要素が最小値より小さい場合、
           最小値を更新する */
        if(test[r] > test[i]) {
            r = i;
        }
    }
    /* 繰り返し終了時の r が最小値となる */
    return r;
}

/* 配列を昇順に並び替える */
void sort_array()
{
    int i, min;
    /* 配列の先頭から繰り返す */
    for(i=0; i<5; i++) {
        /* i 番目以降の最小値を求める */
        min = min_range(i, 4);
        /* i 番目と最長値の場所の値を入れ替える */
        swap_array(i, min);
    }
}

int main(void)
{
    /* show_range() の動作確認 */
    show_range(1, 3);
    show_range(2, 4);
    /* swap_array() の動作確認 */
    swap_array(2, 3);
    swap_array(1, 4);
    show();
    /* min_array() の動作確認 */
    printf("min_array: %d\n", min_array());
    /* min_range() の動作確認 */
    printf("min_range: %d\n", min_range(0, 2));
    printf("min_range: %d\n", min_range(2, 4));
    /* sort_array() の動作確認 */
    sort_array();
    show();

    return 0;
}

```