Ⅰ. compile csd_main.c using ARM instructions

```
#include <stdio.h>

#pragma GCC target ("arm") // compile with ARM instructions

int main(){

    int indata[32] = { 2, 0, -7, -1, 3, 8, -4, 10,
                                         -9, -16, 15, 13, 1, 4, -3, 14,
                                         -8, -10, -15, 6, -13, -5, 9, 12,
                                         -11, -14, -6, 11, 5, 7, -2, -12 };

    int outdata[32];

    int i, j, least, temp;

    size_t size = sizeof(indata) / sizeof(indata[0]); // size of array, "indata"

    for(i = 0; i < size-1; i++) { // sort data in ascending order

        least = i;

        for(j = i+1; j < size; j++) {
            if(indata[j] < indata[least])
                least = j;
        }

        if (i != least) { // if i doesn't stand for the index of the smallest
value
            temp = indata[i];
            indata[i] = indata[least];
            indata[least] = temp; // swap indata[i] and indata[least]
        }
    }

    for(i = 0; i < size; i++){
        outdata[i] = indata[i]; // copy values from indata to outdata
    }

    return 0;

}
```

Ⅱ. compile csd_main.c using Thumb2 instructions

```
#include <stdio.h>

#pragma GCC target ("thumb") // compile with Thumb2 instructions

int main(){

    int indata[32] = { 2, 0, -7, -1, 3, 8, -4, 10,
                       -9, -16, 15, 13, 1, 4, -3, 14,
                       -8, -10, -15, 6, -13, -5, 9, 12,
                       -11, -14, -6, 11, 5, 7, -2, -12 };

    int outdata[32];

    int i, j, least, temp;

    size_t size = sizeof(indata) / sizeof(indata[0]); // size of array, "indata"

    for(i = 0; i < size-1; i++) { // sort data in ascending order

        least = i;

        for(j = i+1; j < size; j++) {
            if(indata[j] < indata[least])
                least = j;
        }

        if (i != least) { // if i doesn't stand for the index of the smallest
value
            temp = indata[i];
            indata[i] = indata[least];
            indata[least] = temp; // swap indata[i] and indata[least]
        }
    }

    for(i = 0; i < size; i++){
        outdata[i] = indata[i]; // copy values from indata to outdata
    }

    return 0;

}
```