```c
#include <stdio.h>

int main(){
    int a[5], b[5];
    int i, j;

    // Note: a[1.0] is illegal access
    // However, a[(int)1.0] is perfectly fine

    // First read all the elements of both arrays
    for(i = 0; i < 5; i++)
        scanf("%d", &a[i]);
    for(i = 0; i < 5; i++)
        scanf("%d", &b[i]);


    // It is a pain to initialize variables such as runningMin
        // Instead of resorting to INFINITY or INT_MAX which can
        // change definitions based on compilers and systems, much
        // better to use a simple flag to state that a value is unset

    // An alternative is of course, to find the maximum value
        // in a, say maxA and set runningMin = maxA + 1 or something
        // similar but this takes more computation. The flag way is cheaper

    int runningMin;
        int isSetRunningMin = 0; // We have not set the value yet
    int isPresent;

    for(i = 0; i < 5; i++){
        // Need to reset flag every time checking a new element of a
        isPresent = 0;
        for(j = 0; j < 5; j++)
            if(a[i] == b[j])
                isPresent = 1;
        // If we find an element of a not present in b
        // Check if it is smaller than the previous such element
        if(!isPresent){
                        if(!isSetRunningMin){ // This is the first element of a not present in b
                                runningMin = a[i]; // No point comparing with runningMin since it
is not set at all
                                isSetRunningMin = 1; // We have now set runningMin to a proper
legal value
                        }else{
                                if(a[i] < runningMin)
                                        runningMin = a[i];
                        }
                }
    }

        if(isSetRunningMin)
                printf("%d", runningMin);
        else
                printf("All elements of a are in b");

    return 0;
}
```