Here is a simple way to represent sets of integers: we store the elements in an array. Suppose we further decide that the elements must be stored in increasing order. Write a function which takes two sets represented in this manner and prints the elements in the intersection of the two in ascending order, one per line. You are to only write the function so that it can be used with the main program below, which has already been entered. For the main() program, you can assume that the elements entered by the user are always in ascending order. Moreover, a set cannot contain any repeated elements.

```cpp
int main(){
    int s1[20], s2[20], n1, n2;
    //Assume that both n1 & n2 are <= 20.
    cin >> n1;
    for(int i=0; i<n1; i++) cin >>s1[i];
    cin >> n2;
    for(int i=0; i<n2; i++) cin >>s2[i];
    // assuming n1,n2 <= 20 and elements given in increasing order
    printIntersection(s1,n1,s2,n2);
}
```

Hint: Adapt the merging procedure seen in the lecture.

**Sample Test Cases**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The due date for submitting this assignment has passed.
As per our records you have not submitted this assignment.

Sample solutions (Provided by instructor)

```
#include <iostream>
#include <cmath>
using namespace std;

void printIntersection(int s1[], int n1, int s2[], int n2){
    int i1 = 0, i2 = 0;
    while(i1<n1 && i2<n2){
        if(s1[i1] == s2[i2]){
            cout << s1[i1] << endl;
            i1++;
            i2++;
        } else if(s1[i1] < s2[i2]) i1++;
        else i2++;
    }
}

int main(){
    int s1[20], s2[20], n1, n2;
    cin >> n1;
    for(int i=0; i<n1; i++) cin >> s1[i];
    cin >> n2;
    for(int i=0; i<n2; i++) cin >> s2[i];
    // assuming n1,n2 <= 20 and elements given in increasing order
    printIntersection(s1,n1,s2,n2);
}
```