

```

#include<iostream>

using namespace std;

class ArrayUtil {
public:
    //s1과 s2를 연결한 새로운 배열을 동적 생성하여 리턴
    static int* concat(int s1[], int size1, int s2[], int size2, int&retSize);
    //s1에서 s2에 있는 숫자를 모두 삭제한 새로운 배열을 동적 생성하여 리턴
    static int* remove(int s1[], int size1, int s2[], int size2, int&retSize);
};

//concat() 구현합니다.
int* ArrayUtil::concat(int s1[], int size1, int s2[], int size2, int&retSize){
    //retSize = size1 + size2;
    //int *p = new int[retSize];
    // p <- s1배열 모두저장
    // p <- s2배열 모두저장
    // return p;
    retSize = size1 + size2;
    int *p = new int[retSize];

    for(int i=0; i<size1; i++){
        p[i] = s1[i];
    }
    for(int i=0; i<size2; i++){
        p[i+size1] = s2[i];
    }
    return p;
}

//remove() 구현합니다.
int* ArrayUtil::remove(int s1[], int size1, int s2[], int size2, int&retSize){
    int *p = new int[size1];

    retSize = size1;

    for(int i=0; i<size1; i++){ // p s1배열 값 저장
        p[i] = s1[i];
    }

    for(int i=0; i<retSize; i++){
        for(int j=0; j<size2; j++){
            if(p[i] == s2[j]){
                for(int k=i; k<retSize; k++){
                    p[k] = p[k+1];
                }
                retSize--;
            }
        }
    }
    return p;
}

```

```

int main() {
    int xcount = 0, ycount = 0;
    int *x, *y;

    //x배열의 원소의 개수(1~5)를 입력 받는다. //입력 받은 수만큼 x배열을 동적 생성한다. //정수를 입력 받아 x배열에 저장한다.
    while(xcount < 1 || xcount > 5){
        cout << "x배열의 원소(1~5)는 몇 개? " << endl;
        cin >> xcount;
    }
    x = new int[xcount];
    cout << "\nx배열에 저장할 정수 입력 : " << endl;
    for(int i=0; i < xcount; i++){
        cin >> x[i];
    }

    //y배열의 원소의 개수(1~5)를 입력 받는다. //입력 받은 수만큼 y배열을 동적 생성한다. //정수를 입력 받아 x배열에 저장한다.
    while(ycount < 1 || ycount > 5){
        cout << "y배열의 원소(1~5)는 몇 개? " << endl;
        cin >> ycount;
    }
    y = new int[ycount];
    cout << "\ny배열에 저장할 정수 입력 : " << endl;
    for(int i=0; i < ycount; i++){
        cin >> y[i];
    }
}

```

```

int retsize = 0;

ArrayUtil A;
//concat() 함수를 호출한다.
// 배열을 통째로 리턴하는건 불가능하기 때문에 동적할당된 배열을 포인터로 받아야한다.
int *conarr = A.concat(x, xcount, y, ycount, retsize);
cout << "\nx배열과 y배열을 연결한 배열은 ";
for (int i = 0; i < retsize; i++) cout << conarr[i] << ' ';
cout << endl;

//remove() 함수를 호출한다.
int *remarr = A.remove(x, xcount, y, ycount, retsize);
cout << "\nx배열에서 y배열의 원소를 삭제한 결과, x배열의 개수는 " << retsize << "개, ";
for (int i = 0; i < retsize; i++) cout << remarr[i] << ' ';
cout << endl;

//동적 할당된 모든 변수를 해제 한다.
delete[] x;
delete[] y;
}

```