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

void riempi(int A1[10][5], int A2[10][5]);
void cross(int A1[10][5],int A2[10][5],bool B[][5]);
bool compareRowColumn(int A1[10][5],int A2[10][5],int rowA1,int columnA2);

main(){
  int A1[10][5],A2[10][5];
  riempi(A1,A2);
  bool B[10][5];
  cross(A1,A2,B);
}

//PRE = Gli array sono di 50 posizioni in totale dentro una matrice da 5x10 elementi
void riempi(int A1[10][5], int A2[10][5]){
    int* pA1 = &A1[0][0];
    for(int i=0;i<50;i++){
        int x;
        cin >> x;
        //A1 righe
        pA1[i] = x;
        //A2 Colonne
        A2[i%10][i/10] = x;
    }

    //Print Output
    //A1
    for(int i=0;i<10;i++){
        for(int e=0;e<5;e++){
            cout << " " <<A1[i][e];
        }
        cout << endl;
    }
    cout <<endl;
    //A2
    for(int i=0;i<10;i++){
        for(int e=0;e<5;e++){
            cout << " " << A2[i][e];
        }
        cout << endl;
    }
    cout << endl;
}
/*
    POST = i due array A1 e A2 sono stati rimepiti con i primi 50 caratteri immessi da tastiera
    Rispettivamente A1 per riga, e A2 epr colonna;
    Alla fine della funzione entrambi gli array vengono stampati a schermo.
*/
```

```
/*
    PRE= gli array A1 e A2 sono matrici di 5x10 elementi ciscuano, tutte gli elementi sono pieni
e
    contengono un input precedentemente immesso da tastiera.
    IN B[i][j], essendo B la matrice presente nella funzione chiamante, sono :
    rowA1 una variabile che contiene l'indice della riga "i" di A1 , con cui cercare se ha qualch
e elemento in comune con la colonna "j" di A2.
    collumnA1 una variabile che contiene l'indice della colonna "j" di A2, con cui cercare se ha
qualche elemento in comune con la riga "i" di A1.
*/
bool compareRowColumn(int A1[10][5],int A2[10][5],int rowA1,int columnA2){
    bool equal = false;
    /**R = ( 0<=cA1<cA1 && !equal ) :
        finche si scorre A2[0..rA2-1][columnA2] &&
        finchè A1[rowA1][0..cA1-1]!=A2[0..rA2-1][columnA2] quindi equal == false
        cioè finchè un elemento della riga rowA1, quale sia esse A1[rowA1][0..cA1-1]
        è != da un elemento della colonna columnA2 quale esso sia A2[0..rA2-1][columnA2]
    */
    for(int cA1=0;cA1<5 && !equal;cA1++){
        /**R = ( 0<=rA2<rA2  && !equal ) :
            finche si scorre A2[0..rA2-1][columnA2] &&
            finchè A1[rowA1][0..cA1-1]!=A2[0..rA2-1][columnA2] quindi equal == false
        **/
        for(int rA2=0;rA2<10 && !equal;rA2++){
            if(A1[rowA1][cA1]==A2[rA2][columnA2]){
                equal=true;
            }
        }//r
    }//c
    /**
       POST alla fine del ciclo, quindi la sua causa di terminazione sarà
       1)o per scorrimento di tutti gli indici dei rispettivi cicli r o c
       2)o per il ritrovamento di un qualsiasi A1[rowA1][0..cA1-1]!=A2[0..rA2-1][columnA2]
       quindi:
       Avremo come parametro di return la variabile equal:
       Se l'insieme dei due cicli finisce per la prima motivazione avremo :
            FALSE perchè
            abbiamo tutti A1[rowA1][0..cA1-1]!=A2[0..rA2-1][columnA2]
       Nel Caso finisse per il secondo motivo avremo
            TRUE perchè
            abbiamo almeno un A1[rowA1][0..cA1-1]==A2[0..rA2-1][columnA2]
    **/
    return equal;
}
/*
    POST=viene returnato dalla funzione :
    TRUE nel caso nella riga rowA1 di A1 è presente un elemento in comune con la collonna columnA
2 di A2;
    FALSE nel caso contrario.
*/
```

```
/**
    PRE=A1 è un array bidimensionale riempito con valori interi di grandezza 10x5 elementi
    A2 è un array bidimensionale riempito con valori interi di grandezza 10x5 elementi
    B è un array bidimensionale vuoto pronto ad ospitare volori booleani con una grandezza di 10x
5 elementi
*/
void cross(int A1[10][5],int A2[10][5],bool B[10][5]){
    /**R = ( 0<=i<50 )
    viene fatto scorrere i da 0 fino a 50 escluso perchè, avendo noi 10 righe per 5 colonne
    facendo 5x10 = 50 posizioni nel array.
    la condizione di permanenza nel ciclo implica solo il scorrere tutti gli elementi del array B
.
    */
    for(int i=0;i<50;i++){
        //Abbiamo come da richiesta B[i][j] con i == i/5 e j == i%5
        //per ogni posizione B[i][j] richiamiamo la funzione compareRowColumn e inseriamo il corr
ispettivo valore booleano.
        B[i/5][i%5]=compareRowColumn(A1,A2,i/5,i%5);
    }
    //POST=Avremo tutti gli elementi di B , una matrice con 5x10 elementi riempita con i corrispe
tivi valori boolean.


    //Print Output
    //B
    for(int i=0;i<10;i++){
        for(int e=0;e<5;e++){
            cout << " " <<B[i][e];
        }
        cout << endl;
    }
}
/**POST= Per ogni elemento di B, viene calcolato il valore del sudetto elemento secondo questa co
ndizione:
 * true sse la riga i di A1 ha qualche elemento in comune con la colonna j di A2
 * con i == al indice delle righe di B e J == al indice delle colonne di B.
 * Alla fine viene stampato per righe l'array B e vengono visualizzati a schermo i dati presenti
 * per ogni posizione B[0..i-1][0..j-1] avremmo :
 * TRUE se
 *      la riga i di A1 ha qualche elemento in comune con la colonna j di A2
 * FALSE
 *      nel caso contrario
*/
```