



שאלה 1 (20 נקודות)

נתונים שלושה מערכים של מספרים שלמים a ו- b , ואוריהם n ו- m בהתאמה. כל מערך ממין בסדר עולה-ממש, וללא חזרות. עליכם לחשב מערך פלט d שיכיל את **החיתוך** של שלושת המערכים; כלומר, ערך כלשהו יופיע ב- d אם ורק אם הוא מופיע בכל שלושת מערכי הקלט. על מערך הפלט d להיות ממין אף הוא בסדר עולה וללא חזרות. כמו כן על הפונקציה להחזיר את מספר האיברים במערך d כערך ההחזרה שלה. לדוגמה, עבור מערכי הקלט:

- a:

1	2	6	9
---	---	---	---
- b:

2	5	6	7	8	9	11
---	---	---	---	---	---	----
- c:

2	6	8	9
---	---	---	---
- d:

2	6	9
---	---	---

יוחזר 3, ויתקבל מערך הפלט הבא:

דרישות סיבוכיות: זמן $O(N)$ (כש- N הוא סכום אורכי שלושת מערכי הקלט), מקום נוסף $O(1)$.

```
int intersect3(int a[], int n, int b[], int m,  
              int c[], int r, int d[]) {
```

```
    int i=0, j=0, k=0, pos=0;
```

```
    while (i<n && j<m && k<r) {
```

```
        if (a[i]==b[j] && b[j]==c[k]) {
```

```
            d[pos++]=a[i++];
```

```
            j++; k++;
```

```
        }
```

```
        else if (a[i]<=b[j] && a[i]<=c[k])
```

```
            i++;
```

```
        else if (b[j]<=a[i] && b[j]<=c[k])
```

```
            j++;
```

```
        else
```

```
            k++;
```

```
    }
```

```
    return pos;
```

```
}
```



פתרון חילופי:

```
int intersect3(int a[], int n, int b[], int m,  
              int c[], int r, int d[]) {
```

```
    int i=0, j=0, k=0, pos=0;
```

```
    while (i<n && j<m && k<r) {
```

```
        int minval = min(a[i],b[j],c[k]), count=0;
```

```
        if (a[i]==minval) { count++; i++; }
```

```
        if (b[j]==minval) { count++; j++; }
```

```
        if (c[k]==minval) { count++; k++; }
```

```
        if (count==3) d[pos++] = a[i-1];
```

```
    }
```

```
    return pos;
```

```
}
```

```
int min(int x, int y, int z) {
```

```
    return min2(min2(x,y),z);
```

```
}
```

```
int min2(int x, int y) {
```

```
    return (x<y ? x : y);
```

```
}
```