

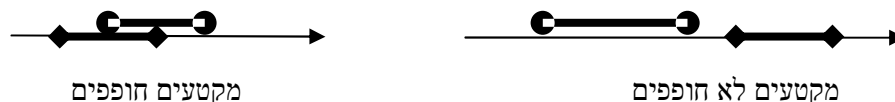


### שאלה 3

נתונים אוסף של מקטעים על ציר  $X$ . כל מקטע מסומן ע"י שני מספרים שלמים  $(x\_start, x\_end)$  המציינים היכן מתחיל המקטע והיכן הוא מסתיים.

#### סעיף א - 15 נקודות

שני מקטעים הם **חופפים** אם החיתוך שלהם אינו ריק. לדוגמא: המקטעים  $(1,3)$  ו  $(2,4)$  חופפים, אך המקטעים  $(3,4)$  ו  $(-1,2)$  אינם חופפים.



כתבו פונקציה שמחשבת את המספר המירבי של מקטעים חופפים עבור קבוצת מקטעים נתונה. הפונקציה מקבלת מערך דו מימדי של מקטעים בשם `segments`, כאשר `segments[0][i]` מכיל את תחילת הסגמנט ה- $i$  ו `segments[1][i]` מכיל את סופו.

לדוגמא, עבור המקטעים:  $(-2,4), (3,7), (5,10), (6,8), (9,14)$  המספר המירבי של מקטעים חופפים הוא 3, מכיוון שאת הקטע  $(6, 7)$  מכסים 3 מקטעים.

#### הנחות והנחיות נוספות:

- הניחו שמספר המקטעים מוגדר ב `#define N`.
- **חשוב:** הניחו שבכל נקודה מתחיל או מסתיים לכל היותר מקטע אחד.
- מותר להשתמש בפונקציה `msort(int arr[], int n)` המפעילה את אלגוריתם merge sort לצורך מיון מערך באורך  $n$ . מותר לשנות את תוכן המערך `segments`.
- הקפידו על סיבוכיות זמן ומקום נוסף קטנים ככל האפשר (אם יש סתירה בין השתיים – סיבוכיות זמן עדיפה על סיבוכיות מקום). פתרון נכון שאינו עומד בדרישה יזכה לניקוד חלקי.

```
int overlapping_segments(int segments[][N])
{
    int start = 0, end = 0, overlapping = 0, maxo = 0;
    msort(&(segments[0][0]), N);
    msort(&(segments[1][0]), N);

    while(start < N) {
        if(segments[0][start] < segments[1][end]) {
            start++;
            overlapping++;
        }
        else { // segments[0][start] > segments[1][end]
            end++;
            overlapping--;
        }
    }
}
```

```

    }
    maxo = (overlapping > maxo) ? overlapping : maxo;
}
return maxo;

```



## סעיף ב - 10 נקודות

נקודה תיקרא **מכוסה ע"י מקטע** אם קיים מקטע המכיל אותה. למשל המקטע  $[1,10]$  מכיל את הנקודה 7 אך אינו מכיל את הנקודה 11. אם הנקודה ממוקמת על גבול המקטע היא נחשבת מוכלת בו.

בסעיף זה נניח כי כל המקטעים הם באורך  $k$ .

כתבו פונקציה המקבלת מערך חד מימדי של התחלות מקטעים כאלו בשם **segments**, את אורך המקטעים  $k$  וכן נקודה  $x$ , ומחזירה 1 אם הנקודה  $x$  מכוסה ע"י מקטע כלשהו או אפס אחרת. שימו לב שהמערך ממוין לפי נקודות ההתחלה, כלומר מקטע המתחיל לפני מקטע אחר יופיע לפניו. לדוגמא, עבור המקטעים המתחילים ב-10 וב-15  $\text{segments}[] = \{10, 15\}$  ו- $k=3$ :

- עבור  $x=12$  תחזיר הפונקציה 1
- עבור  $x=14$  תחזיר הפונקציה 0

### הנחות והנחיות נוספות:

- הניחו שמספר המקטעים מוגדר ב  $\text{\#define } N$ .
- הניחו שאין שני מקטעים צמודים, כלומר שאין נקודה המשמשת כנקודת התחלה או סיום של שני סגמנטים שונים.
- הקפידו על סיבוכיות זמן ומקום נוסף קטנים ככל האפשר. פתרון נכון שאינו עומד בדרישה יזכה לניקוד חלקי בלבד.

```
int is_covered(int arr[N], int k, int x)
{
    int start = 0, end = N-1, curr;

    while (start <= end) {
        curr = (start + end)/2;
        if (x < arr[curr])
            end = curr-1;
        else if (x <= arr[curr]+k)
            return 1;
        else
            start = curr+1;
    }
    return 0;
}
```