

## שאלה 5 (20 נקודות)

נגדיר מבנה כדלקמן המתאר נקודה במישור הקואורדינטות

```
struct point {  
    double x, y;  
};  
  
typedef struct point Point;
```

הגדרה מערך מספרים דו-ממדי  $a[M][N]$  נקרא ממין בעמודות אם לכל  $0 \leq i < M-1$  ולכל  $0 \leq j < N$  מתקיים  $a[i][j] \leq a[i+1][j]$  (דהיינו כל האיברים באותה העמודה מסודרים בסדר לא יורד).

הגדרה מערך מספרים דו-ממדי  $a[M][N]$  נקרא ממין בשורות אם לכל  $0 \leq i < M$  ולכל  $0 \leq j < N-1$  מתקיים  $a[i][j] \leq a[i][j+1]$  (דהיינו כל האיברים באותה השורה מסודרים בסדר לא יורד).

נתון מערך דו-ממדי  $a[M][N]$  Point כאשר הקבועים  $M$  ו- $N$  הם חזקות של 2 ומוגדרים באמצעות `#define`. מטרת השאלה היא לכתוב פונקציה בשפת C אשר ממיינת את המערך כך שקואורדינטות  $x$  של איבריו יהיו ממוינות בשורות וקואורדינטות  $y$  של איבריו יהיו ממוינות בעמודות.

דוגמא המערך  $a[3][3]$  הבא עונה על דרישות השאלה:

$a[0][0] = (1,1)$	$a[0][1] = (2,5)$	$a[0][2] = (9,1)$
$a[1][0] = (1,2)$	$a[1][1] = (4,6)$	$a[1][2] = (4,2)$
$a[2][0] = (1,7)$	$a[2][1] = (3,9)$	$a[2][2] = (8,7)$

(כאשר סדר הקואורדינטות בדוגמא הוא  $(x,y)$ ).

### סעיף א

השלם את המקומות החסרים בפונקציה `merge_by_x` אשר ממזגת שני מערכים חד-ממדים מטיפוס Point הממוינים בסדר לא יורד לפי ערך של קואורדינטה  $x$ .

```
void merge_by_x (Point a[], int na, Point b[], int nb,  
Point c[])  
{  
    int ia, ib, ic;  
    for(ia = ib = ic = 0; (ia < na)&&(ib < nb); ic++)  
    {  
        if( _____ ) {  
            c[ic] = a[ia];  
            ia++;  
        }  
        else {  
            _____;  
            ib++;  
        }  
    }  
    for(; ia < na; ia++, ic++) _____;  
    for( _____ ) _____;  
}
```

## סעיף ב

השלם את המקומות החסרים בפונקציות merge\_sort\_by\_x (ופונקצית עזר רקורסיבית internal\_msort\_by\_x) אשר ממיינת מערך חד-ממדי מטיפוס Point בסדר לא יורד לפי ערך של קואורדינטה x.

```
void merge_sort_by_x (Point a[], int n)
```

```
{
    _____ tmp_array;
    tmp_array = (_____)malloc(____);
    internal_msort_by_x(a, n, tmp_array);
    free(tmp_array);
}
```

```
void internal_msort_by_x (_____ a[], _____ n, _____ tmp_array[])
```

```
{
    int left, right;
    if (n < 2) return;
    left = n / 2;
    right = n - left;
    internal_msort_by_x (_____, _____, _____);
    internal_msort_by_x (_____, _____, _____);
    merge_by_x (_____, _____, _____, _____, _____);
    memcpy(a, tmp_array, _____);
}
```

### סעיף ג

השלם את הפונקציה `sort_by_x_and_y` אשר ממיינת את המערך כך שקואורדינטות  $x$  של איבריו תהיינה ממוינות בשורות וקואורדינטות  $y$  של איבריו תהיינה ממוינות בעמודות. הפונקציה תעבוד בסיבוכיות זמן  $O(MN(\log M + \log N))$ .

הפונקציה יכולה להיעזר (אך לא חייבת להשתמש בכולם) ב:

- פונקציות `malloc` ו-`memcpy`.
- פונקציות מסעיפים א'-ב' הממזגות וממיינות מערך חד-ממדי לפי ערכים של קואורדינטות  $x$ .
- פונקציות אשר דומות לפונקציות מסעיפים א'-ב', אך ממזגות וממיינות מערך חד-ממדי לפי ערכים של קואורדינטות  $y$ . החתימות של הפונקציות הן:

```
void merge_by_y (Point a[], int na, Point b[], int nb, Point c[])
void merge_sort_by_y (Point a[], int n)
```

### הערה

שימו לב שבמערך דו ממדי כלשהו - לדוגמה `a[10][10]`, ניתן לשלוח את השורה `a[3]` (או כל שורה אחרת) לפונקציה שמצפה לקבל מערך חד ממדי עם הגודל המתאים של המערך למשל `f(a[3],10)`.

### פתרון:

```
void sort_by_x_and_y (Point a[M][N])
{
```

```
    merge_sort_by_y( &a[0][0], N*M );
```

```
}
```

הסבר את הפתרון שלך. מדוע הפונקציה שכתבת עומדת בדרישות הסיבוכיות?