

שאלה 6 (20 נקודות)

שאלה זו (על כל סעיפיה) מתייחסת להגדרות הבאות:

```
#define NROWS (10)
```

```
#define NCOLS (50)
```

```
typedef enum { ADD, SUBTRACT } plusminus_t;
```

חלק א (3 נקודות)

השלימו את הפונקציה הבאה, המקבלת מערך חד מימדי `arr` מטיפוס `unsigned int` ואת אורכו `len`, ומחזירה את הערך הנמוך ביותר במערך.

```
unsigned int min_value(unsigned int arr[], unsigned int len)
```

```
{
    _____
    _____
    _____
    _____
}
```

חלק ב (3 נקודות)

השלימו את הפונקציה הבאה, המקבלת ארבעה פרמטרים:

- מערך חד מימדי `arr` מטיפוס `unsigned int`.
- מערך חד מימדי `sum` מטיפוס `int`.
- אורך `len` (שני המערכים `arr` ו-`sum` הינם מאורך זה).
- ערך `operation` מטיפוס `plusminus_t`.

אם `operation` הינו `ADD`, על הפונקציה לחבר למערך `sum` את המערך `arr` (חיבור איבר-איבר).
 אם `operation` הינו `SUBTRACT`, על הפונקציה להחסיר מהמערך `sum` את המערך `arr` (חיסור איבר-איבר).
 בשני המקרים, הכוונה היא לעדכן כל כניסה במערך `sum` בהתאם לערך הכניסה המקבילה במערך `arr` ולערך הפרמטר `operation`.

```
void update_sum ( unsigned int arr[],
                  int          sum[],
                  unsigned int len,
                  plusminus_t op )
```

```
{
    _____
    _____
    _____
    _____
}
```

חלק ג (14 נקודות)

בסעיף זה עליכם יהיה להשלים את הפונקציה `covering_rows`, המקבלת:
 1. מערך דו-מימדי מטיפוס `unsigned int` שבו `NROWS` שורות ו- `NCOLS` עמודות.

2. מספר שלם `setsize`
 ומדפיסה את כל הקבוצות של `setsize` שורות בדיוק כך שמתקיים התנאי הבא:
 סכום `setsize` השורות (איבר-איבר, כחיבור וקטורים) אינו מכיל אפס.

דוגמה: עבור המטריצה הבאה ו- `setsize=3`

השורה ה- 0
 השורה ה- 1
 השורה ה- 2
 השורה ה- 3

1	3	0	1	0	1	0
1	8	1	0	0	0	1
0	7	0	1	1	0	1
0	5	0	0	0	1	1

על הפלט להיות שתי השורות הבאות (סדר השורות אינו משנה):

0,1,2

1,2,3

2	18	1	2	1	1	2
---	----	---	---	---	---	---

כי הסכום (איבר-איבר) של שלוש השורות הראשונות הוא

1	20	1	1	1	1	3
---	----	---	---	---	---	---

סכומן (איבר-איבר) של שלוש השורות האחרונות הוא

וסכום של כל קבוצה אחרת של שלוש שורות מכיל אפס בלפחות אחת מהעמודות.

הדרכה:

- בפונקציה `covering_rows` הגדירו (ואתחלו) שטחי עזר.
- הפונקציה `covering_rows` נעזרת בפונקציה `covering_help` – השלימו גם אותה.
- העזרו בפונקציות שכתבתם בסעיפים הקודמים של שאלה זו, ובפונקצית ההדפסה הבאה:

```
void print_positive ( unsigned int arr[], int len)
{
    int k;

    for ( k = 0; k < len; k++)
        if ( arr[k] ) printf ("%d ", k) ;
    printf("\n");
}
```

```
void covering_rows ( unsigned int matrix[][NCOLS], int setsize )
{
    unsigned int help_mem1[_____] = {0};

    int help_mem2[_____] = {0};

    covering_help ( matrix, setsize, _____, _____, _____ );
    return;
}
```

עמוד 13

```

void covering_help ( unsigned int      matrix[][NCOLS],
                    int               setsize,
                    unsigned int * subset,
                    int               * sum,
                    _____ )
{
    /*******/

    if ( !setsize )
    {
        if ( _____ )

            print_positive(_____);

        return;
    }

    if ( setsize + _____ > _____ ) return;

    /*******/

    subset [ _____ ] = _____;

    update_sum ( _____, _____, _____, _____ );

    covering_help (matrix, _____, _____, _____, _____ );

    /*******/

    _____;

    _____;

    _____;

    return;
}

```

