

מבוא למדעי המחשב
מועד א', סמסטר א' תשס"ב, 7/2/02

מרצה: שולי וינטנר.
מתרגלים: שלמה יונה, ליאת לונטל.

משך המבחן: שתיים וחצי.
חומר עזר: מותר כל חומר עזר, מלבד מחשב.
הנחיות:

1. ודאו כי בטופס שבידיכם 7 עמודים. יש לכתוב את התשובות על גבי טופס המבחן ולהגיש את כל הטופס ואת הטופס בלבד.
2. קראו היטב כל שאלה. ודאו כי אתם מבינים את השאלה לפני שתתחילו לענות עליה.
3. כתבו בכתב יד ברור וקריא. השתמשו בדפי הטיוטה והעתיקו לטופס המבחן רק תשובות סופיות. תשובות לא קריאות לא תיבדקנה.
4. הערות לתשובותיכם ניתן לכתוב בעברית, גם בגוף פונקציות C.
5. אם לא נכתב אחרת, כאשר עליכם להגדיר פונקציה יש להגדיר פונקציה אחת בדיוק. לא ניתן להשתמש בפונקציות חיצוניות.
6. אם לא נכתב אחרת, בתוכניות ניתן להשתמש בפונקציות מתוך הספריות הבאות בלבד:
 - a. `stdio.h`
 - b. `stdlib.h`
 - c. `string.h`
 - d. `ctype.h`

בהצלחה!

שאלה	ציון
1	/20
2	/20
3	/25
4	/25
5	/10
סה"כ	/100



שאלה 1-20 נקודות:

נתון מערך המוגדר כך: `int array[N]`. `N` הוא קבוע המוגדר ב-`#define`.
ידוע כי המערך מכיל את כל המספרים השלמים בין 0 ל-`N`, פרט לאחד.
כתבו פונקציה המקבלת מערך כזה ומחזירה את המספר החסר בו.
לדוגמה, אם `N=5` והמערך הוא:

0	1	5	3	4
---	---	---	---	---

, על הפונקציה להחזיר 2.
על הפונקציה לעבוד בזמן $O(N)$. פתרונות בסיבוכיות גבוהה יותר לא יתקבלו.

```
int missing_number (int array[])
{
    int temp[N+1];
    int i;

    for (i=0; i<N+1; i++)
        temp[i] = 0; /* i does not occur in array */
    for (i=0; i<N+1; i++)
        temp[array[i]]++; /* array[i] occurs in array */
    for (i=0; i<N+1; i++)
        if (! temp[i])
            return i;
}
```



שאלה 2-20 נקודות:

בשאלה זו ניתן להשתמש בכל הפונקציות שהודגמו בהרצאה, ללא צורך להגדיר אותן. אם הנכם משתמשים בפונקציה חיצונית כזו, הצהירו עליה והסבירו בהערה מה היא מבצעת.
 נתון מערך של $2N$ מספרים שלמים המוגדר כך: `int array[N*2]`.
 כתבו פונקציה המקבלת מערך כזה (ואת גודלו) ומחלקת את אברי המערך לשתי קבוצות בגודל N , A ו- B , כך שההפרש בין סכום אברי A לסכום אברי B יהיה מקסימאלי.
 על הפונקציה להדפיס את אברי A ואחריהם את אברי B , עם הודעות מתאימות. סדר ההדפסה אינו חשוב.

לדוגמה: אם $N=2$, והמערך הוא:

54	9	2	21
----	---	---	----

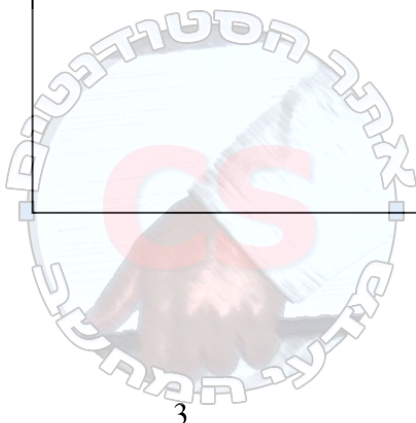
 : הפונקציה תדפיס:
 A's members are: 9,
 B's members are: 54, 21

על הפונקציה לעבוד בזמן $O(N \log N)$. פתרונות בסיבוכיות גבוהה יותר לא יתקבלו.

```
void mergesort (int array[], int n); /* sort an array. O(n log n) */

void print_groups (int array[], int n)
{
    int i;

    mergesort (array, n);
    printf ("Group A: ");
    for (i=0; i<n/2; i++)
        printf ("%d ", array[i]);
    printf ("\nGroup B: ");
    for (i=n/2; i<n; i++)
        printf ("%d ", array[i]);
    printf ("\n");
}
```



שאלה 3-25 נקודות:

נייצג תמונה בשחור לבן ע"י מערך דו-מימדי, שכל תא בו מכיל 1 עבור נקודה שחורה, ו-0 עבור נקודה לבנה. נתונה ההגדרה:

```
typedef enum {WHITE, BLACK} Pixel;
typedef Pixel Picture[N][M];
```

כאשר N ו-M מוגדרים ב-#define.

כתבו פונקציה בשם num_black_pixels המקבלת תמונה ומחזירה את מספר הנקודות השחורות בה.

```
int num_black_pixels (Picture p)
{
    int i,j, num=0;

    for (i=0; i<N; i++)
        for (j=0; j<M; j++)
            if (p[i][j])
                num++;
    printf ("Picture has %d black pixels\n", num);
    return num;
}
```

כתבו פונקציה המקבלת תמונה ומחזירה 1 אם התמונה היא של קו ישר בעובי 1, ו-0 אחרת. קו ישר יכול להיות קו מאונך, מאוזן או אלכסוני בשיפוע של 45° . נקודה בודדת היא קו. לדוגמה: עבור התמונה הבאה: $N=5, M=6$

0	0	0	0	0	0
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	1	0	0
0	0	0	0	0	0

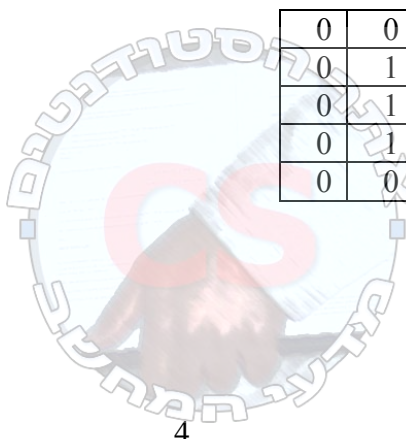
הפונקציה תחזיר 1.

0	0	0	0	0	0
0	1	1	1	1	0
0	1	0	0	0	0
0	1	0	0	0	0
0	0	0	0	0	0

ועבור התמונה הבאה:

הפונקציה תחזיר 0.

ניתן להשתמש בפונקציה num_black_pixels גם אם לא הגדרתם אותה.



```

int is_line (Picture p)
{
    int i,j,k,l;
    int done = 0;

    for (i=0; i<N && !done; i++)
        for (j=0; j<M && !done; j++)
            if (p[i][j])
                done = 1;
    j--; i--;
    if (!done) /* no BLACK pixels */
        return 0;
    printf ("found first BLACK pixel at (%d,%d)\n", i, j);
    /* check horizontal */
    for (k=j+1; k<M && p[i][k]; k++)
        ;
    if (k>j+1) { /* horizontal pixels found */
        printf ("horizontal line found from (%d,%d) to (%d,%d)\n",
            i,j,i,k-1);
        return (num_black_pixels (p) == (k-j));
    }
    /* check vertical */
    for (k=i+1; k<N && p[k][j]; k++)
        ;
    if (k>i+1) { /* vertical pixels found */
        printf ("vertical line found from (%d,%d) to (%d,%d)\n",
            i,j,k-1,j);
        return (num_black_pixels (p) == (k-i));
    }
    /* check first diagonal */
    for (k=i+1, l=j+1; k<N && l<M && p[k][l]; k++,l++)
        ;
    if (k>i+1) { /* diagonal pixels found */
        printf ("diagonal line found from (%d,%d) to (%d,%d)\n",
            i,j,k-1,l-1);
        return (num_black_pixels (p) == (k-i));
    }
    /* check second diagonal */
    for (k=i+1, l=j-1; k<N && l>=0 && p[k][l]; k++,l--)
        ;
    if (k>i+1) { /* diagonal pixels found */
        printf ("diagonal line found from (%d,%d) to (%d,%d)\n",
            i,j,k-1,l+1);
        return (num_black_pixels (p) == (k-i));
    }
    return 1; /* a single pixel is a line */
}
  
```

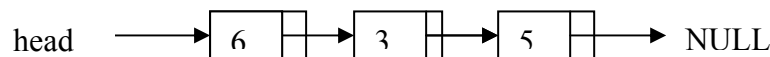
שאלה 4- 25 נקודות:

נתונה רשימה מקושרת אשר כל צומת בה מוגדר כך:

```
typedef struct node {
    int contents;
    struct node *next;
} Node;
```

ידוע כי ברשימה לפחות איבר אחד, וכי האיבר האחרון ברשימה מצביע ל-NULL. הפונקציה **הרקורסיבית** `print_half` מקבלת מצביע לתחילת רשימה מקושרת, ומדפיסה, **בסדר הפוך**, את האיברים הבאים:

- אם מספר האיברים ברשימה זוגי, יודפסו האיברים במקומות הזוגיים ברשימה;
 - אם מספר האיברים ברשימה אי-זוגי, יודפסו האיברים במקומות האי-זוגיים.
- לדוגמה: אם הרשימה היא הרשימה הבאה:



לאחר הקריאה לפונקציה יהיה הפלט:

5 6

השלימו את הגדרת הפונקציה. בכל מקום ריק יש לכתוב ביטוי אחד בדיוק.

```
int print_half (Node *list)
{
    int toprint;

    if (list == NULL) {
        return TRUE;
    } else {
        toprint = print_half (list->next)
        if (toprint) {
            printf ("%d ", list->contents);
        }
        return !toprint;
    }
}
```



שאלה 5-10 נקודות:

השלימו את הפלט במקומות המתאימים:

```
#include <stdio.h>
int a=0;

void f1(int *p, int q) {
    a = (*p)+q;
    (*p) /= 10;
    q -= (*p);
}

void f2(int *p, int *q) {
    if ((*p % 10) < (*q % 10)) {
        f1(p,*q);
    }
}

int main( ) {
    int b,c,i;
    a=2000;
    b=2001;
    c=2002;
    f1(&a,b);
    printf("a=%d  b=%d\n", a, b);

    {
        int a=10;
        b=20;
        f1(&a,b);
        printf("a=%d  b=%d\n", a, b);
    }

    {
        int a[ ]={111,222,333};
        for (i=0;i<2;i++){
            f2(a+i,a+i+1);
        }
        printf("a[0]=%d a[1]=%d  a[2]=%d\n",
            a[0], a[1], a[2]);
    }

    return 0;
}
```

a= 400 b= 2001

a= 1 b= 20

a[0]= 11 a[1]= 22 a[2]= 333

