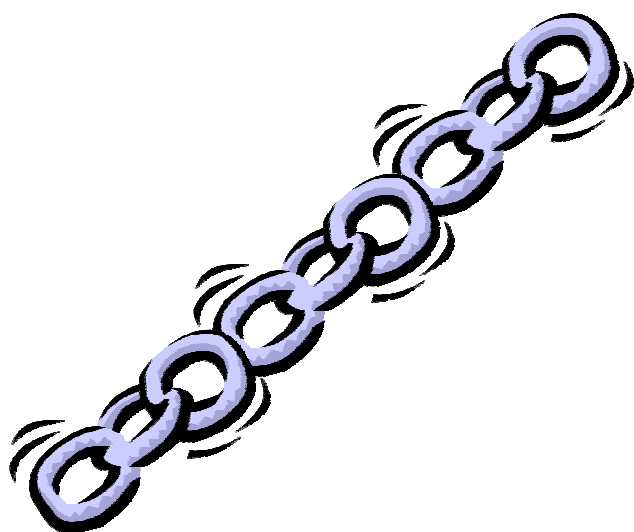


4. מחלקת רשימה



התבניות שבמחלקה

תבניות בניה

4.1 - בניה של רשימה

4.2 - שרשור רשימות

תבניות סריקה

4.3 - נמצא ברשימה?

4.4 - מקום ברשימה

4.5 - מחיקת איברים מרשימה

עבודה עם סמן

4.6 - העברות ברשימה

4.7 - היפוך רשימה

מבנה הנתונים רשימה הוא אחד ממבני הנתונים הבסיסיים ביותר. רעיון השימוש במבנה זה הוא שמירת נתונים בצורה סדרתית כך שלכל איבר, מלבד האיברים הראשון והאחרון, ישנו איבר עוקב ואיבר קודם. באיבר נשמר נתון אחד או יותר, וישנו מצביע המצביע על האיבר הבא ברשימה. הנתון (נתונים) והמצביע נשמרים בדרך כלל ברשומה. ערך המצביע באיבר האחרון ברשימה הוא ערך המצייין "סוף רשימה".

רשימה היא בעצם מבנה נתונים אלטרנטיבי למבנה הנתונים מערך. ייחודו של מבנה הנתונים מערך הוא בכך שניתן לגשת ישירות לכל איבר בו, באמצעות מצייין האיבר, ואין צורך לשמור באיבר את המצביע לאיבר הבא. לעומת זאת, ברשימה לא ניתן לגשת ישירות לאיבר כלשהו. יש לגשת לעוגן "ולעבור" דרך הרשימה עד ההגעה לאיבר הרצוי. מכאן, שבמבנה הנתונים מערך נוח ויעיל הרבה יותר לגשת לאיבר.

לעומת זאת, פעולת ההכנסה של איבר נוחה יותר ברשימה. הכנסה של איבר ברשימה היא פעולה מיידיית אשר דורשת עידכון של שני מצביעים בלבד (מצביע אל האיבר המוכנס, מן האיבר שלפניו ברשימה, ומצביע מן האיבר המוכנס לאיבר שאחריו). במערך גורמת הפעולה הכנסה להזזת כל האיברים שאחרי האיבר המוכנס, ואינה פעולה מיידיית.

פעולת ההוצאה של איבר לפעמים גם נוחה יותר ברשימה מאשר במערך. זאת אם בעת זימון הפעולה יינתן לא רק מצביע לאיבר המוצא אלא גם מצביע לאיבר שלפניו. ישנה אפשרות להשתמש ב"רשימה דו-כיוונית", אשר בה לכל איבר ישנם שני מצביעים – מצביע לאיבר הבא ומצביע לאיבר הקודם. עבור רשימה כזו, פעולת הוצאה היא חסכונית גם כאשר ניתן בעת הזימון רק המצביע לאיבר המוצא. יחד עם זאת, שימוש ברשימה כזו הינו מורכב יחסית, ובדרך כלל לא נדון בו.

בספר הלימוד המקורי של "עיצוב תכנה" מוצג מערך כאמצעי למימוש של רשימה. בחומר לימוד המובא כאן נשתדל לדון באפשרות זו כמה שפחות. זאת כיון שאנו רואים מערך כמבנה נתונים אלטרנטיבי לרשימה, ולא כמבנה נתונים למימוש רשימה. רוב ספרי הלימוד אודות מבני נתונים במדעי המחשב מציגים מימוש רשימה באמצעות הקצאת זיכרון דינמית, תוך שימוש במצביעים ("פוינטרים") מתאימים. מימוש זה מביא לידי ביטוי בצורה מתאימה את הרעיון של הפעולות המבוצעות ברשימה. ישנם ספרים שמציגים מימוש רשימה באמצעות מערך, אך בהם אין בהכרח סידור של איברי הרשימה באופן סדרתי לפי איברי המערך, אלא סידור דינמי, המשתנה בהתאם לפעולות ההכנסה והוצאה של איברים מן הרשימה. אנו לא נדון במימוש זה בחומר הלימוד הנוכחי.

שימוש ברשימה מתאים במקרים בהם כמות הנתונים שצריך לשמור בצורה סדרתית עשויה להיות גדולה, וגודלה לא ידוע מראש. בנוסף, מתאים להשתמש ברשימה כאשר צפוי שבמהלך עיבוד הנתונים יגיעו בקשות חוזרות ונשנות של הכנסה והוצאה של איברים. במקרים כאלה נבחר ברשימה, ולא במערך, כיון שעיבוד באמצעות רשימה איננו מחייב אותנו לקבוע מראש חסם של כמות הנתונים, ומאפשר לנו פעולות הכנסה והוצאה "נוחות" ופחות "יקרות". במקרים בהם ניתן לדעת ולקבוע מראש את כמות הנתונים, שאיננה גדולה מדי, ועיקר הפעולות על הנתונים הן פעולות איחזור (ולא הכנסה/הוצאה) נעדיף שימוש במערך על פני שימוש ברשימה.

לאור המתואר לעיל, אנו מתייחסים לרשימה ברמה המופשטת כאל שרשרת חוליות. החוליה הראשונה ברשימה היא חוליה מיוחדת המבטאת את תחילת הרשימה, והיא עוגן הרשימה. בנוסף לכך, אנו מניחים מספר הנחות בשימוש ברשימה: 1. רשימה ריקה היא רשימה אשר מכילה "עוגן-רשימה" ו-"סוף רשימה" בלבד; 2. פעולת איחזור מרשימה איננה מוגדרת על רשימה ריקה; 3. רשימה המועברת כפרמטר היא רשימה שאותחלה; היא יכולה להיות רשימה ריקה.

הזכרנו קודם שנתונים השמורים באיבר ברשימה נשמרים בדרך כלל ברשומה. ישנם מקרים בהם מדובר בנתון בודד, וישנם מקרים בהם מדובר במספר נתונים. בפרק זה איננו מבחינים בין מקרים בהם מדובר בנתון בודד לבין מקרים בהם מדובר במספר נתונים. אנו מכנים בדרך-כלל את הנתון/נתונים השמורים באיבר בשם x. אנו מכנים בדרך כלל את ערכו של מצביע לאיבר (לרשומה) בשם p.

יחידות ספריה לטיפוס הנתונים **רשימה** נמצאות באתר תבניות : www.tau.ac.il/~csedu

4.1 - בניה של רשימה

נקודת מוצא: סדרה של ערכים.

מטרה: בניית רשימה שמכילה את סדרת הערכים.

אלגוריתם: בנה-רשימה

אתחל-רשימה $L \leftarrow$

$p \leftarrow$ (L) רשימה

כל זיו לא סוף-הצורה ג'ד

הנתון הבא הצורה $x \leftarrow$

הכנס-רשימה (L, p, x)

דוק-רשימה $p \leftarrow (L, p)$

חזר את L

הערות

- תבנית זו היא התבנית הראשונית של יצירת, או בניית רשימה. ביישומים רבים נבנית תחילה רשימה, ולאחר מכן מתבצעות על הרשימה פעולות שונות. ערכי סדרת הערכים הניתנים כקלט משמשים לבנייה ומהווים את ערכי איברי הרשימה – כל ערך בסדרת נתוני הקלט נשמר באיבר אחר ברשימה.
- לפני קליטת הנתונים מאותחל המצביע p , כך שיצביע על עוגן הרשימה. כל תוספת של איבר היא לסוף הרשימה, לאחר האיבר שעליו מצביע p . בעקבות תוספת של איבר מוצב p על האיבר האחרון, שהוסף. הצבה זו נעשית בהוראה האחרונה בלולאה. לו הוראה זו היתה מושמטת, p היה מוצב לכל אורך מהלך הביצוע על עוגן הרשימה, והיתה נוצרת רשימה שסידורה (נתוני האיברים שבה) הפוך לסדר הנתונים בסדרת הקלט.

Pascal

(* פעולה המחזירה רשימה המכילה סדרה של ערכים *)

```
procedure list_build (var L: list_type);
var
    p: pos_type;
    x: list_info_type;
begin
    list_init (L);
    p := list_anchor (L);
    get_next_element (x);      { פעולה המחזירה את האיבר הבא ברשימה }
    while x <> end_of_input do
        begin
            list_insert (L, p, x);
            p := list_next (L, p);
            get_next_element (x);
        end;
    end;
end;
```

C

/* פעולה המחזירה רשימה המכילה סדרה של ערכים */

```
list_type list_build (void)
{
    list_type L;
    pos_type p;
    list_info_type x;

    L = list_init ();
    p = list_anchor (L);
    x = get_next_element ();    // פעולה המחזירה את האיבר הבא ברשימה
    while (x != end_of_input)
    {
        list_insert (L, p, x );
        p = list_next (L, p);
        x = get_next_element ();
    }
    return L;
}
```

שאלות

שאלה 4.1.1

- א. פתח אלגוריתם שהקלט שלו הוא סדרת מספרים ממויינים, ומטרתו היא בניית רשימה כך שכל נתון קלט שמור באיבר אחר ברשימה.
- ב. פתח אלגוריתם שמייצר סדרת מספרים אקראיים בסדר עולה. הפלט יהיה רשימה המורכבת ממספרים אלה.
- ג. ממש בסביבת העבודה את האלגוריתם שכתבת בכל אחד מהסעיפים הקודמים.

■

שאלה 4.1.2

- א. פתח אלגוריתם שיקלוט ויחזיר רשימה של המועמדים בתחרות כשרונות צעירים.
- ב. ממש בסביבת העבודה את האלגוריתם שכתבת בסעיף הקודם.

■

שאלה 4.1.3

- א. פתח אלגוריתם שיקלוט סדרה של מספרים בתחום 0-9 ויצור מהם רשימה שהיא פלינדרום, כלומר: הפלט שיתקבל מהדפסת איברי הרשימה מההתחלה לסוף יהיה זהה לפלט שיתקבל מהדפסת הרשימה מהסוף להתחלה.

לדוגמה: עבור הקלט: $2, 5, 4$

תתקבל הרשימה: L: 2, 5, 4, 4, 5, 2

- ב. ממש בסביבת העבודה את האלגוריתם שכתבת בסעיף הקודם.
- ג. מהי סיבוכיות האלגוריתם שכתבת? נמק.

■

שאלה 4.1.4

טיפוס הנתונים **ציונו-של-תלמיד** מוגדר כך:

Type	Pascal	typedef struct	C
Grade_type = record stud_num : longint; grade : integer; end;		{ long stud_num; int grade; } grade_type;	

- בממשק לטיפוס נתונים זה קיימת גם הפעולה: **אתחל-ציון** (מספר-תלמיד, ציון). פעולה זו מקבלת את מספר התלמיד ואת ציונו ומחזירה רשומת ציון מאותחלת.
- פתח אלגוריתם שיקלוט פרטי תלמידים, ועבור כל תלמיד יאתחל את ציונו-של-תלמיד ויחזיר רשימה של ציוני התלמידים בכיתה.

■

שאלה 4.1.5

פתח אלגוריתם שיקלוט למשתנה n את מספר הצלעות של מצולע ואחריו סידרה של n קואורדינטות (x,y) ממשיים, ויחזיר רשימה שכל אחד מאיבריה יהיה נקודה (מטיפוס `point_type`) המהווה קודקוד במצולע. טיפוס הנתונים `point_type` הינו רשומה המכילה מצייני x ו- y במישור. x ו- y מטיפוס ממשי.

■

שאלה 4.1.6

מפאת הביקוש הרב לסעוד ב"מסעדה של רחל" יש להזמין מקומות מראש. פתח אלגוריתם שיקלוט הזמנות מלקוחות ויחזיר את רשימת ההזמנות למסעדה. פרטי ההזמנה יכללו את שם המזמין, טלפון להתקשרות, שעת הביקור המבוקשת במסעדה ומספר הסועדים המתוכנן.

■

שאלה 4.1.7

בבית הספר החלו לארגן את מסיבת הסיום של תלמידי י"ב. התלמידים המעוניינים להשתתף במופע התבקשו למלא טופס בו ירשמו את שמם, מספר הטלפון שלהם והתחום בו הם יכולים לתרום – משחק, נגינה, שירה או ריקוד.

א. פתח אלגוריתם שיקלוט את הטפסים ויחזיר רשימה של נתוני התלמידים המעוניינים להופיע.
ב. פתח אלגוריתם שיקבל כפרמטר את רשימת המעוניינים מהסעיף הקודם, ויחזיר שתי רשימות - האחת תכלול את פרטי התלמידים שביקשו לתרום בשירה ו/או בנגינה והשניה את פרטי התלמידים שביקשו לתרום במשחק ו/או ריקוד.

■

שאלה 4.1.8

פתח אלגוריתם שיקבל 2 רשימות ממויינות $L1$ ו- $L2$ של מספרים שלמים ללא חזרות (כל מספר מופיע פעם אחת בלבד ברשימה). האלגוריתם יחזיר רשימה $L3$ שהיא רשימה ממויינת המכילה את כל המספרים שב- $L1$ וב- $L2$ (רשימת המיזוג). אם איבר מופיע בשתי הרשימות, יש לכלול אותו פעם אחת בלבד ברשימת המיזוג.

■

4.2 - שרשור רשימות

נקודת מוצא: שתי רשימות מאותחלות.

מטרה: שרשור עותק של הרשימה השנייה לסופה של הרשימה הראשונה.

אלגוריתם: שרשר-רשימות (L1, L2)

קוצם-הרשימה (ל1-רשימה) (L1, L1) ← p1

עוקם-הרשימה (ע1-רשימה) (L2, L2) ← p2

כא ע1 ≠ p2 ע1-רשימה (L2) ע2

אחץ-הרשימה (L2, p2) ← x

הכנס-לרשימה (L1, p1, x)

עוקם-הרשימה (L1, p1) ← p1

עוקם-הרשימה (L2, p2) ← p2

הערות

- שרשור של שתי רשימות הוא בעצם הכנסת רשימה אחת, L2 בסופה של רשימה אחרת, L1. שרשור נחוץ במקרים רבים, כאשר יש לצרף רשימת נתונים אשר נצברה במקום אחד לרשימת נתונים אשר נצברה במקום נפרד.
- אנו מתייחסים לשרשור באופן הבא: לאחר השרשור, הרשימה L2 נותרת רשימה נפרדת מ-L1, שהופכת להיות רשימה ארוכה יותר, שחלקה השני זהה ל-L2 מבחינת ערכי האיברים. כלומר, השרשור הינו בעצם העתקה של L2 לסופה של L1. המשמעות של העתקה זו היא ששינוי עתידי ב-L2 לא ישפיע על L1, ושינוי עתידי ב-L1 לא ישפיע על L2.
- משמעות ההעתקה של L2 לסופה של L1 היא שעבור כל איבר ב-L2 נבנה איבר חדש, בעל תוכן זהה ומוכנס במקום המתאים ב-L1, כפי שמתבצע בלולאה המוצגת בתבנית.

- ישנן צורות נוספות, מלבד העתקה, שבהן ניתן להתייחס לשרשור. בפרט, ניתן להתייחס לשרשור כמצב שבו L2 הופכת להיות חלק מ-L1 (למעשה "הזנב" של L1), ואינה נותרת כרשימה נפרדת. במקרה כזה, כל שינוי עתידי ב-L2 פירושו גם שינוי ב-L1. מקרה כזה מחייב משנה זהירות בכל שינוי באחת הרשימות לאחר שרשורן. מקרה זה הינו מעבר לחומר הלימוד של היחידה "עיצוב תכנה", הן עקב משנה הזהירות הנדרש בו והן עקב היבטים שונים במורכבות מימושו (כגון עיבוד מתאים של פעולות עם עוגן הרשימה L2).
- ההנחה בנקודת המוצא היא ששתי הרשימות מאותחלות ולא בהכרח לא ריקות. בתחילת הביצוע מחושב המצביע על האיבר שלפני סוף הרשימה L1. אם הרשימה לא ריקה, זהו האיבר האחרון ב-L1 ואם הרשימה ריקה, זהו עוגן הרשימה L1. כמו כן מחושב המצביע על האיבר הראשון ב-L2. אם L2 לא ריקה, זהו האיבר הראשון ב-L2, ואם הרשימה ריקה, זהו סוף הרשימה. לאחר חישובים אלה מתבצעת בלולאת התבנית ההעתקה של L2 לסופה של L1.

(* פעולה המקבלת שתי רשימות מאותחלות ומשרשרת את הרשימה השניה *)
 (* לסופה של הרשימה הראשונה (מבלי לקלקל את הרשימה השניה). *)

```

procedure list_concat (var L1: list_type; L2: list_type);
var
    p1, p2 : pos_type;
    x : list_info_type;
begin
    p1 := list_prev (L1, list_end (L1));
    p2 := list_next (L2, list_anchor (L2));
    while p2 <> list_end (L2) do
        begin
            list_retrieve (L2, p2, x);
            list_insert (L1, p1, x);
            p1 := list_next (L1, p1);
            p2 := list_next (L2, p2);
        end ;
    end;

```

/* פעולה המקבלת שתי רשימות מאותחלות ומשרשרת את הרשימה השניה */
 /* לסופה של הרשימה הראשונה (מבלי לקלקל את הרשימה השניה). */

```

void list_concat (list_type L1, list_type L2)
{
    pos_type p1, p2;
    list_info_type x;

    p1 = list_prev (L1, list_end (L1));
    p2 = list_next (L2, list_anchor (L2));
    while (p2 != list_end (L2))
    {
        x = list_retrieve (L2, p2);
        list_insert (L1, p1, x);
        p1 = list_next (L1, p1);
        p2 = list_next (L2, p2);
    }
}

```

שאלות

שאלה 4.2.1

- א. איזה שינוי יש לבצע באלגוריתם "שרשר-רשימות" אם נחליף את הפעולה **אחזר-מרשימה** (L2, p2) שבתוך הלולאה בפעולה **הוצא-מרשימה** (L2, p2)? נמק.
- ב. כיצד תראה הרשימה המשורשרת אם נמחק את הפעולה **עוקב-ברשימה** (L1, p1) מהלולאה?
- ג. מהי סיבוכיות הפעולה **שרשר-רשימות** כפי שהיא מוצגת בתבנית? נמק.
-

שאלה 4.2.2

- נתונות שתי קבוצות לימוד. כל קבוצה מורכבת מרשימה של תלמידים. ידוע שאין תלמיד הרשום לשתי קבוצות הלימוד. בגלל מיעוט הנרשמים, הוחלט לאחד אותן לקבוצת לימוד אחת. פתח אלגוריתם שיוסיף את תלמידי הקבוצה השנייה לסוף הקבוצה הראשונה.
-

שאלה 4.2.3

- עקב מיעוט הנרשמים לטיסה מסוימת, החליטה חברת התעופה להציע לנוסעים להמיר את מועד טיסתם המקורי לטיסה זו בתמורה להנחה במחיר הכרטיס.
- א. פתח אלגוריתם שיקבל כפרמטר את רשימת הנוסעים שהסכימו להמרת כרטיסי הטיסה שלהם, ואת רשימת הנוסעים המקורית וישרשר את רשימת הנוסעים שהסכימו להמרה לסוף רשימת הנוסעים המקורית.
- ב. ממש בסביבת העבודה את האלגוריתם שכתבת בסעיף הקודם.
-

4.3 - נמצא ברשימה

נקודת מוצא: רשימה מאותחלת, ערך x .
מטרה: פעולה בוליאנית הבודקת האם הערך y נמצא באחד מאיברי רשימה L .

אלגוריתם: נמצא-ברשימה? (L, x)

$L \leftarrow (L, (L) \text{ - } \text{פונקציה-רשימה})$

כל $p \neq \text{פונקציה-רשימה}(L)$ $\neq p$ $\neq \text{פונקציה-רשימה}(L)$

$y \leftarrow (L, p)$ אחזר-מפונקציה

אם x $\neq \text{פונקציה-רשימה}(L)$ $\neq y$ $\neq \text{פונקציה-רשימה}(L)$

החזר "אמת"

$p \leftarrow (L, p)$ פונקציה-רשימה

החזר "שקר"

הערות

- ישנן מערכות תכנה רבות אשר בהן נשמרים נתונים ברשימות, ולפעמים יש לברר אם נתון מסויים שמור באחד האיברים של רשימה. הפעולה המתוארת בתבנית, של בדיקה האם באחד מאיברי הרשימה שמור ערך מבוקש, מיועדת לביצוע בירור מן הסוג המתואר.
- הבדיקה אודות הימצאות ערך מבוקש היא למעשה חיפוש הערך באיברי הרשימה, תוך סריקת האיברים לפי הסדר, החל מן הראשון. במידה והערך נמצא, מסתיים החיפוש מייד ומוחזרת התשובה "אמת", אחרת מתנהל החיפוש עד ההגעה לסוף הרשימה ומוחזר הערך "שקר". שים לב שלא מוחזר מקומו של האיבר שבו נמצא הערך המבוקש. חיפוש אשר בעקבותיו מוחזר מקום האיבר מופיע בתבנית הבאה, תבנית 4.4.
- ייתכן שערך מבוקש יופיע במספר איברים ברשימה, והמטרה לא תהיה לדעת האם הערך נמצא או לא-נמצא ברשימה, אלא לדעת מהו מספר האיברים בהם נמצא הערך המבוקש (התשובה 0 תוחזר כאשר הערך לא נמצא). במקרה כזה יש לשנות במקצת את התבנית, כך שתבצע מנייה מתאימה במקום חיפוש. המנייה תחייב בכל מקרה מעבר על כל איברי הרשימה.

(* פעולה המקבלת רשימה L ואיבר x ומחזירה true אם x נמצא ברשימה ו-false אחרת. הנחות: הרשימה L מאותחלת. x הוא איבר מטיפוס איברי הרשימה. *)

```
function list_find_x (L: list_type; x: list_info_type): boolean;
var
  p : pos_type;
  y : list_info_type;
  found : boolean;
begin
  found := false;
  p := list_next (L, list_anchor (L));
  while (p <> list_end (L)) and not found do
    begin
      list_retrieve (L, p, y);
      if x = y then found := true;
      p := List_next (L, p);
    end ;
  list_find_x := found;
end;
```

```
#define TRUE 1
#define FALSE 0

/* פעולה המקבלת רשימה L ואיבר x ומחזירה true אם x נמצא ברשימה ו-false אחרת. הנחות: הרשימה L מאותחלת. x הוא איבר מטיפוס איברי הרשימה. */

int list_find_x (list_type L, list_info_type x)
{
  pos_type p;
  list_info_type y;

  p = list_next (L, list_anchor (L));
  while (p != list_end (L))
  {
    y = list_retrieve (L, p);
    if (x == y) return TRUE;
    p = list_next (L, p);
  }
  return FALSE;
}
```

שאלות

שאלה 4.3.1

לאחר מבחני הקבלה הארציים ללשכת הקוסמים, ביקש מחנך הכיתה לדעת כמה מתלמידיו עברו את הבחינה בהצלחה.

- א. פתח אלגוריתם שיקבל כפרמטר שתי רשימות - הראשונה: רשימת שמות העוברים את הבחינה הארצית, השנייה: רשימת תלמידי הכיתה. האלגוריתם יחזיר את מספר תלמידי הכיתה שהצליחו בבחינה.
- ב. ממש בסביבת העבודה את האלגוריתם שכתבת בסעיף הקודם.

■

שאלה 4.3.2

לפני בית המשפט התבררה תלונה לפיה לשכת הקוסמים מכשילה בכוונת תחילה מועמדים חדשים, ולראייה אף מועמד לא קיבל ציון גבוה מ-60 בבחינות הארציות. מועצת הלשכה טענה כי רוב הנבחנים מקבלים ציונים טובים מאוד. על מנת לבחון את נכונות הטענה, הציעו השופטים להגריל ציון בתחום 80 - 100 (כולל) ולבדוק האם קיים לפחות נבחן אחד שקיבל ציון שערכו גבוה או שווה לציון זה.

פתח אלגוריתם שיקבל כפרמטר את רשימת ציוני הנבחנים ואת הציון שהוגרל, ויחזיר 'אמת' אם קיים נבחן כזה ברשימה ו'שקר' אחרת.

■

שאלה 4.3.3

הפעולה **מיד-אחרי**? (L, x, y) מקבלת כפרמטר רשימה L לא ריקה וערכים x ו- y מטיפוס איברי הרשימה. הפעולה תחזיר 'אמת' אם הערך x מופיע ברשימה מיד אחרי המופע של y או הערך y מופיע ברשימה מיד אחרי המופע של x , ו'שקר' אחרת.

א. פתח אלגוריתם שיממש את הפעולה **מיד-אחרי**? (L, x, y).

ב. מה סיבוכיות זמן הריצה של האלגוריתם שכתבת? נמק.

■

שאלה 4.3.4

פתח אלגוריתם בשם רשימת-ריבועים ($L1, L2$) שיקבל כפרמטר שתי רשימות $L1$ ו- $L2$ המכילות מספרים שלמים, ויחזיר רשימה שלישית שתכיל את כל איברי $L1$ שהערך הריבועי שלהם נמצא ב- $L2$. לדוגמה: אם 3 נמצא ב- $L1$, אזי הוא יהיה ברשימה החדשה רק אם 9 נמצא ב- $L2$.

יש לפתח שני אלגוריתם לפי כל אחד מהתנאים הבאים:

א. הרשימות $L1$ ו- $L2$ ממויינות.

ב. הרשימות $L1$ ו- $L2$ אינן ממויינות.

■

שאלה 4.3.5

נתונות שתי רשימות L1 ו-L2, לא ממויינות.

- א. פתח אלגוריתם בשם **חיתוך-רשימות** (L1, L2) שיקבל את שתי הרשימות L1 ו-L2 ויחזיר רשימה שתכיל את כל האיברים הנמצאים בשתי הרשימות גם יחד.
- ב. פתח אלגוריתם בשם **רשימת-המשלים** (L1, L2) שיקבל את שתי הרשימות L1 ו-L2 ויחזיר רשימה שתכיל את כל איברי L1 שאינם ב-L2.
- ג. מה סיבוכיות האלגוריתמים **חיתוך-רשימות** ו**רשימת-המשלים** שכתבת? נמק.

■

שאלה 4.3.6

במחשב בית הספר משובצים N התלמידים לקבוצות לימוד שונות. כל תלמיד יכול להופיע ביותר מקבוצת לימוד אחת. לכל קבוצת לימוד יש מורה. מורה יכול ללמד יותר מקבוצת לימוד אחת.

- א. פתח אלגוריתם שיקבל מערך בגודל k של קבוצות הלימוד המשוייכות למורה מסוים (כל איבר במערך הוא רשימה של תלמידים המהווה קבוצת לימוד). האלגוריתם יחזיר את רשימת התלמידים הלומדים אצל המורה. אם תלמיד מופיע ביותר מקבוצת לימוד אחת, יופיע שמו ברשימה החדשה פעם אחת בלבד.
- ב. אם N הוא החסם העליון למספר התלמידים בקבוצת לימוד, מהי סיבוכיות הפעולה שכתבת? נמק. שים לב: מספר התלמידים בקבוצת הלימוד ה- i הוא n_i .
- ג. מה יש לשנות בפעולה שכתבת אם קבוצות הלימוד של מורה יישמרו ברשימה של רשימות ולא במערך?

■

4.4 - מקום ברשימה

נקודת מוצא: רשימה מאותחלת, ערך x .

מטרה: מציאת מקומו של האיבר הראשון ברשימה אשר בו נמצא ערך y המקיים תנאי ביחס לערך x שניתן כפרמטר.

אלגוריתם: מקום-ברשימה (L, x)

$p \leftarrow (L, (L, x))$ *פונקציה-מקום-ברשימה*

$\text{כאשר } p \neq \text{פונקציה-מקום-ברשימה}(L, x)$

$y \leftarrow (L, p)$ *מחזר-מקום-ברשימה*

אם y מקיים-תנאי-לבדיקה ביחס f של x אזי

החזר את p

$p \leftarrow (L, p)$ *פונקציה-מקום-ברשימה*

החזר את p

הערות

- במקרה הפשוט ביותר, משמעות הניסוח (שבאלגוריתם) " y מקיים-תנאי-לבדיקה ביחס ל- x " פירושו " y שווה ל- x ". במקרה מורכב יותר יתכן שנרצה לבדוק אם y שווה לפונקציה כלשהי של x , שאיננה זהות (למשל, y גדול מ- x).
- תבנית זו דומה מאד לתבנית הקודמת, תבנית 4.3, אלא שכאן מוחזר מצביע לאיבר שעבורו מתקיים התנאי, ואילו בתבנית הקודמת הוחזר הערך "אמת", שמשמעותו שקיים ברשימה איבר שערכו מקיים התנאי הנדרש. השימוש בכל אחת משתי התבניות נגזר מן הבעיה שיש לפתור. למשל, כאשר בכל איבר ברשימה נשמרים נתונים על תלמיד אחר בכתה, ומעוניינים לדעת אם בכיתה ישנו תלמיד שגובהו עולה על 1.90 מ', אזי מספיק להשתמש בתבנית הקודמת. אם מעוניינים גם לדעת, למשל את שמו של התלמיד, אזי יש להשתמש בתבנית הנוכחית.
- יתכן שברשימה מספר איברים שעבורם מתקיים התנאי הדרוש. האלגוריתם המוצג מחזיר מצביע לראשון מביניהם. ניתן לשנות את האלגוריתם, כך שיחזיר למשל את האחרון שבהם.
- בדומה להרחבה שצוינה בהערות התבנית הקודמת, גם כאן יתכן שנהיה מעוניינים במצייני כל האיברים שעבורם מתקיים התנאי הנבדק. במקרה כזה, יחזיר האלגוריתם מצביע על רשימה של האיברים האלה, או על רשימה של מקומותיהם ברשימה L . רשימה זו יש לבנות תוך כדי סריקת איברי L , ושימוש בתבנית הבניה 4.1. כתיבת האלגוריתם להרחבה זו הינה מורכבת, וכוללת שילוב של תבניות (4.4 ו-4.1).

<p>(* פעולה המקבלת רשימה L וערך x ומחזירה את מקומו של האיבר הראשון ברשימה שערכו שווה לערך x. הנחות: הרשימה L מאותחלת, x הוא מטיפוס הערכים ברשימה. *)</p> <pre> function list_find_place (L: list_type; x: list_info_type): pos_type; var p: pos_type; y: list_info_type; found: boolean; begin found := false; p := list_next (L, list_anchor (L)); while (p <> list_end (L)) and not found do begin list_retrieve (L, p, y); if x = y then found := true else p := list_next (L, p); end; list_find_place := p; end;</pre>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">Pascal</div>
---	---

<p>/* פעולה המקבלת רשימה L וערך x ומחזירה את מקומו של האיבר הראשון ברשימה שערכו שווה לערך x. הנחות: הרשימה L מאותחלת, x הוא מטיפוס הערכים ברשימה. */</p> <pre> pos_type list_find_place (list_type L, list_info_type x) { pos_type p; list_info_type y; p = list_next (L, list_anchor (L)); while (p != list_end (L)) { y = list_retrieve (L, p); if (x == y) return p; p = list_next (L, p); } return p; }</pre>	<div style="border: 1px solid black; padding: 5px; display: inline-block;">C</div>
---	--

שאלות

שאלה 4.4.1

פתח אלגוריתם שיקבל כפרמטר רשימה L המכילה את הטמפרטורות היומיות שנרשמו בחודש מסויים באזור המרכז ואת הטמפרטורה החודשית הממוצעת לאזור זה, ויחזיר את היום הראשון בחודש שבו נרשמה טמפרטורה מעל ממוצע זה.

■

שאלה 4.4.2

פתח אלגוריתם שיקבל כפרמטר את רשימת הציונים של תלמידי הכיתה.
א. האלגוריתם יחזיר את מקומו של הציון הראשון הגבוה מהממוצע.
ב. האלגוריתם יחזיר את מקומו של הציון המקסימלי בכיתה.

■

שאלה 4.4.3

בספריית וידיאו מוחזקים פרטי הלקוחות ברשימה. לקוח המבקש לשאול סרט, מוסר את מספר המנוי שלו ופרטיו מופעים על מסך התצוגה.
פתח אלגוריתם שיקבל כפרמטר את רשימת הלקוחות ומספר המנוי ויחזיר את פרטי הלקוח.

■

שאלה 4.4.4

לצורך הקמת נבחרת שחייה, נערכו מבחני קבלה. תוצאות המבחנים של כל השחיינים נשמרו ברשימה. בתום המבחנים הוחלט לקבל לנבחרת כל שחיין שתוצאת המבחן שלו גבוהה מסף מסוים.
א. פתח אלגוריתם שיקבל כפרמטר את רשימת התוצאות ואת תוצאת הסף, ויחזיר רשימה חדשה המכילה כל השחיינים שהשיגו תוצאה הגבוהה מסף קבלה זה.
ב. ממש בסביבת העבודה את האלגוריתם שכתבת בסעיף הקודם.
ג. אם ידוע שמספר השחיינים ברשימה הוא N , מהי סיבוכיות האלגוריתם שכתבת? נמק.

■

שאלה 4.4.5

המורה ניסים החליט למחשב את ציוני תלמידיו. ברשותו שתי חבילות כרטיסים, כרטיס לכל תלמיד: בחבילה הראשונה כולל כל כרטיס: מספר תלמיד וציונו הממוצע בבחנים. בחבילה השנייה כולל כל כרטיס: מספר תלמיד וציונו הממוצע במבחנים.
לכל תלמיד קיים כרטיס אחד ויחיד בכל אחת מהחבילות. חבילות הכרטיסים אינן ממויינות. ציונו הסופי של תלמיד נקבע כממוצע משוקלל של הציונים הממוצעים במבחנים ובבחנים: משקל 70% למבחנים ומשקל 30% לבחנים.
א. פתח אלגוריתם שיקבל כפרמטר שתי רשימות שכל איבר בהן הוא מטיפוס כרטיס-ציון. אלגוריתם יחזיר רשימה חדשה שתכיל את הציונים הסופיים של תלמידי המורה ניסים.
ב. מהי סיבוכיות האלגוריתם שכתבת? נמק.

■

שאלה 4.4.6

פתח אלגוריתם, יעיל ככל שתוכל, שיקבל כפרמטר רשימה של מספרים שלמים, לא ממויינים וערך x ויחזיר את מקומו של האיבר האחרון ברשימה שערכו x .

■

שאלה 4.4.7

פתח אלגוריתם, יעיל ככל שתוכל, שיקבל כפרמטר רשימה של מספרים שלמים, ויחזיר את המקום הראשון בו נמצאו שני מספרים זהים העוקבים זה לזה.
דוגמה:

L: 1, 8, 3, 5, 5, 4, 7, 7, 9, 2 ברשימה L יוחזר המקום:
p ↗

■

שאלה 4.4.8

פתח אלגוריתם, יעיל ככל האפשר, שיקבל כפרמטר שתי רשימות L1 ו-L2 של מספרים שלמים, ויחזיר את מקומה של תחילת רשימה L2 בתוך L1. אם L1 לא מכילה את L2, יוחזר המקום סוף הרשימה ב-L1.
דוגמה:

L1: 2, 3, 1, 3, 7, 6, 9, 4, 8 עבור הרשימות:
L2: 3, 7, 6
L1: 2, 3, 1, 3, 7, 6, 9, 4, 8 יוחזר המקום:
p ↗

■

4.5 - מחיקת איברים מרשימה

נקודת מוצא: רשימה מאותחלת, ערך x .

מטרה: מחיקת כל המופעים של x ברשימה L .

אלגוריתם: מחק-מרשימה (L, x)

$p \leftarrow (L, L)$ מחיקת-מרשימה (L, x)

כל p \neq x מחיקת-מרשימה (L, x)

$x =$ מחיקת-מרשימה (L, p), L

מחיקת-מרשימה (L, p)

אחרת

$p \leftarrow (L, p)$ מחיקת-מרשימה

הערות

- תבנית זו מציגה זימון חוזר של הפעולה הוצא-מרשימה, כאשר עבור כל אחד מהאיברים המוצאים מתקיים תנאי נדרש (כאן, שוויון בין ערכו לערך הנתון x). האלגוריתם מבצע סריקה של הרשימה מתחילתה לסופה. ציינו במבוא לפרק זה, על תבניות של רשימה, שביצוע הפעולה "הוצא מרשימה" יכול להיות "יקר יחסית" או "זול", מבחינת יעילות, בהתאם לסוג הרשימה ו/או לפרמטרים הניתנים בעת הזימון של פעולת ההוצאה.
- במידה והרשימה היא רשימה "רגילה", בה לכל איבר מצביע אחד על האיבר הבא, הרי זימון כפי שמוצג בתבנית כאן, הינו "יקר" יחסית, כיון שהוצאת איבר מן הרשימה מחייבת סריקה של הרשימה מתחילתה, כדי להגיע למצביע על האיבר שלפני האיבר המוצא. ולכן, אמנם התבנית מציגה אלגוריתם אלגנטי, אך אלגוריתם זה איננו יעיל במיוחד. האלגוריתם יהיה יעיל אם הרשימה היא רשימה דו-כיוונית, ובה לכל איבר שני מצביעים – מצביע "קדימה" ומצביע "אחורה" (ראה מבוא לפרק). אפשרות אחרת היא לאפשר יעילות, תוך שימוש ברשימה "רגילה", על-ידי זימון של פעולת הוצאה המקבלת שני מצביעים – מצביע לאיבר המוצא ומצביע לאיבר שלפניו. במקרה כזה ישונה האלגוריתם שבתבנית, לאלגוריתם שבו תבצע סריקה של הרשימה L , מתחילתה עד סופה, עם שני מצביעים p_1 ו- p_2 , אשר אחד מהם מצביע על האיבר הנוכחי (אליו הגיעה הסריקה) והשני מצביע על קודמו.
- מקרה מיוחד של מחיקת איברים מרשימה יכול להיות מחיקת רצף של איברים (בתחילת הרשימה, בסופה, או באמצעה). ההחלטה על האיברים למחיקה לא תהיה בדרך-כלל על-פי השוואת ערכו של כל איבר לערך x נתון, אלא על-פי תנאים מסוימים שיוגדרו (למשל, אם איברי הרשימה שומרים ציונים, המסודרים בסדר עולה, תנאי יכול להיות "כל איבר שהציון בו מעל 80", במקרה כזה יש למחוק תת-רשימה רציפה של איברים, החל ממקום מסוים ברשימה המקורית ועד סופה). במקרים כגון המקרה

המתואר, ישונה האלגוריתם, כך ששרשרת זימוני ההוצאה של איברים רצופים תבצע בפשטות וביעילות.

<pre> (* פעולה המקבלת רשימה L ואיבר x ומוחקת את כל המופעים של x מהרשימה. הנחות: הרשימה L מאותחלת. x הוא איבר מטיפוס איברי הרשימה. *) procedure list_delete_x (var L: list_type; x: list_info_type); var p : pos_type; begin p := list_next (L, list_anchor (L)); while p <> list_end (L) do begin list_retrieve (L, p, y); if x = y then list_delete (L, p) else p := list_next (L, p); end; end; end; </pre>	<div style="border: 1px solid black; padding: 5px; display: inline-block;"><i>Pascal</i></div>
---	--

<pre> /* פעולה המקבלת רשימה L ואיבר x ומוחקת את כל המופעים של x מהרשימה. הנחות: הרשימה L מאותחלת. x הוא איבר מטיפוס איברי הרשימה. */ void list_delete_x (list_type L, list_info_type x) { pos_type p; p = list_next (L, list_anchor (L)); while (p != list_end (L)) { if (list_retrieve (L, p) == x) list_delete (L, &p); else p = list_next (L, p); } } </pre>	<div style="border: 1px solid black; padding: 5px; display: inline-block;"><i>C</i></div>
---	---

שאלות

שאלה 4.5.1

- א. פתח אלגוריתם שיקבל כפרמטר רשימה של מספרים ומספר x , וימחק מהרשימה את כל המופעים שערכם שווה לפעמיים x .
- ב. ממש בסביבת העבודה את האלגוריתם שכתבת בסעיף הקודם.

■

שאלה 4.5.2

- א. פתח אלגוריתם שיקבל כפרמטר רשימה ממויינת של מספרים. המספרים יכולים לחזור על עצמם יותר מפעם אחת. האלגוריתם יחזיר את הרשימה ללא כפילויות.
- ב. פתח אלגוריתם המקבל רשימה לא ממויינת, ומבצע אותה פעולה כמו בסעיף הקודם.

■

שאלה 4.5.3

במגמת מדעי הטבע היה ביקוש רב למסלול "השבחת בנות". הוחלט לפצל את הקבוצה לשלוש קבוצות לימוד. פתח אלגוריתם שיקבל כפרמטר את רשימת הנרשמים למסלול זה, ויעביר את כל התלמידים שביקשו להתמחות ב"כיפוף בנות" לרשימה אחת וכל אלו שביקשו להתמחות ב"יישור בנות" לרשימה אחרת. כל האחרים יישארו ברשימה המקורית.

■

שאלה 4.5.4

נתונות שתי רשימות $L1$ ו- $L2$, כך ש- $L2$ מוכלת בתוך $L1$. יש למחוק את $L2$ מתוך $L1$.
דוגמה:

הרשימות לפני המחיקה: $L1: 5, 3, 7, 3, 7, 2, 9, 1, 8$

$L2: 3, 7, 2$

$L1$ אחרי המחיקה: $L1: 5, 3, 7, 9, 1, 8$

■

שאלה 4.5.5

- נתונות שתי רשימות $L1$ ו- $L2$. כל רשימה מכילה איברים השונים זה מזה (רשימה ללא חזרות). פתח אלגוריתם שיקבל כפרמטר את הרשימות וימחק מ- $L1$ את כל האיברים המופיעים ב- $L2$, עבור כל אחד מהמקרים הבאים:
- א. שתי הרשימות ממויינות.
- ב. שתי הרשימות אינן ממויינות.

■

4.6 - העברות ברשימה

נקודת מוצא: רשימה מאותחלת.

מטרה: העברת כל האיברים שערכיהם מקיימים תנאי לתחילת הרשימה, תוך שמירה על הסדר היחסי של האיברים המועברים.

אלגוריתם: סידור-רשימה (L)

$pos \leftarrow (L)$ *על-רשימה (L)*

$p \leftarrow (L, pos)$ *על-רשימה (L, pos)*

$\delta \neq p$ *כאשר $\delta \neq p$ (L) רשימה*

$x \leftarrow (L, p)$ *אחרי-רשימה (L, p)*

אם x מקיים תנאי הבדיקה אזי

(L, p) *רשימה-אחרי (L, p)*

(L, pos, x) *רשימה-אחרי (L, pos, x)*

$pos \leftarrow (L, pos)$ *על-רשימה (L, pos)*

אחרי

$p \leftarrow (L, p)$ *על-רשימה (L, p)*

הערות

- ביישומי תכנה שונים יש לעיתים צורך לעדכן רשימה על-ידי העברת איברים ממקומם למקום אחר ברשימה. דוגמה שכיחה למקרים כאלה היא "סידור", או "סידור חלקי" של איברי הרשימה על פי ערכים השמורים בהם. למשל, ריכוז בתחילת הרשימה של כל האיברים שבהם שמורים משקלי תלמידים ששוקלים מעל 100 ק"ג. במקרים כאלה יש לעבור על הרשימה מתחילתה, ולהעביר כל איבר המקיים תנאי דרוש למקום אחר ברשימה (כגון תחילתה).
- התבנית המוצגת כאן מתארת סדרה של העברות לתחילת הרשימה, בהתאם לקיום תנאי. התנאי המתואר כאן, בצורה פשוטנית, הוא יחס כלשהו בין הנתון x , לבין הערכים השמורים באיברי הרשימה. המאפיין המרכזי שיש לשים אליו לב בתבנית זו הוא השימוש ב**שני** מצביעים שונים – מצביע אחד לאיבר אליו הגיע מהלך הסריקה של הרשימה L עד כה, שהוא האיבר הבא שיש לבחון את הצורך

בהעברתו, ומצביע שני למקום ברשימה אליו יועבר האיבר הבא שיועבר ממקומו המקורי ב-L. שני המצביעים נקראים כאן ו-p. pos.

- צורת ההתקדמות עם שני המצביעים המתוארים איננה משנה את הסדר היחסי בין איברי הרשימה הלא-מועברים. כמו כן, הסדר היחסי בין האיברים המועברים איננו משתנה. כלומר, אם איבר א' נמצא ב-L לפני איבר ב', לפני ביצוע ההעברות, ושניהם מועברים במהלך ההעברות, אזי גם לאחר ההעברות יימצא איבר א' לפני איבר ב'. למעשה, ניתן להתבונן בסדרת ההעברות כעל יצירה של תת-רשימה תחילית של L, אשר מורכבת מאיברים נבחרים מתוך L, שהועברו לתחילתה, תוך שמירת הסדר היחסי ביניהם.
- גם כאשר נרצה להעביר תת-קבוצה של איברים לסוף הרשימה, או למקום אחר בה, שאיננו תחילתה, יהיה האלגוריתם מבוסס על שימוש בשני מצביעים, בדומה לדרך המוצגת כאן. התבנית של שימוש בשני מצביעים שאחד מהם מצביע על המקום הבא בהתקדמות הסדרתית לאורך הרשימה, והשני מצביע על המקום אליו יש להעביר את האיבר הנבחר הבא, היא תבנית שימושית בהקשרים מגוונים בעיצוב תכנה.

Pascal

(* פעולה המקבלת רשימה L ומעבירה לתחילת הרשימה את כל המספרים השליליים, תוך שמירה על סדר הופעתם ברשימה המקורית. הנחה: הרשימה L מאותחלת. *)

```
procedure list_shift (var L: list_type );
var
    pos, p : pos_type;
    x : list_info_type;
begin
    pos := list_anchor (L);
    p := list_next (L, pos);
    while p <> list_end (L) do
        begin
            list_retrieve (L, p, x);
            if x < 0 then
                begin
                    list_delete (L, p);
                    list_insert (L, pos, x);
                    pos := list_next (L, pos);
                end
            else
                p := list_next (L, p);
            end;
        end;
end;
```

```
/* פעולה המקבלת רשימה L ומעבירה לתחילת הרשימה את כל המספרים
   השליליים, תוך שמירה על סדר הופעתם ברשימה המקורית.
   הנחה: הרשימה L מאותחלת. */

void list_shift (list_type L)
{
    pos_type pos, p;
    list_info_type x;

    pos = list_anchor (L);
    p = list_next (L, pos);

    while (p != list_end (L))
    {
        x = list_retrieve (L, p);
        if (x < 0)
        {
            list_delete (L, &p);
            list_insert (L, pos, x);
            pos = list_next (L, pos);
        }
        else
            p = list_next (L, p);
    }
}
```

שאלות

שאלה 4.6.1

פתח אלגוריתם שיקבל כפרמטר רשימה L המכילה מספרים שלמים. על האלגוריתם להעביר לתחילת הרשימה את כל המספרים החיוביים תוך שמירה על הסדר היחסי של כל איבר ברשימה, ולהשאיר את הלא-חיוביים בסוף הרשימה (המספרים הלא חיוביים הם 0 והמספרים השליליים).

■

שאלה 4.6.2

בית הספר קבע כי תלמיד שציונו השנתי במדעי המחשב הוא 85 ומעלה, יוכל להמשיך ללימודי "עיצוב תכנה". תלמיד מיוצג ברשומה הכוללת את שמו ואת ציונו השנתי. פתח אלגוריתם שיקבל כפרמטר רשימה של תלמידים וציוניהם השנתיים, הרשימה ממויינת עפ"י א"ב שמות התלמידים. על האלגוריתם להעביר לתחילת הרשימה את התלמידים היכולים להמשיך בלימודי "עיצוב תכנה", תוך שמירת המיון האלפביתי.

■

שאלה 4.6.3

פתח אלגוריתם שיקבל כפרמטר רשימה של פרטי תלמידים הכוללים שם ומגמה מועדפת. קיימות שלוש מגמות אפשריות – תכנון, פיתוח ועיצוב. הרשימה ממויינת לפי א"ב שמות התלמידים. על האלגוריתם להעביר את כל התלמידים שבחרו בתכנון לתחילת הרשימה ואת התלמידים שבחרו בעיצוב לסוף הרשימה. תלמידי הפיתוח יישארו באמצע הרשימה. אין לפגוע במיון הפנימי בכל אחד משלושת חלקי הרשימה.

■

שאלה 4.6.4

פתח שלושה אלגוריתמים שכל אחד מהם יקבל כפרמטר רשימה L לא ממויינת, של מספרים, ויחזיר את הרשימה ממויינת בכל אחת מהדרכים הבאות:

א. בשיטת מיון-בועות.

ב. בשיטת מיון-בחירה.

ג. בשיטת מיון-הכנסה.

■

שאלה 4.6.5

נתון האלגוריתם הבא :

מה-עושה? (L)

{ הרשימה L היא רשימה של אפסים ואחדים המיוצגת באמצעות שרשרת חוליות.
 הנחות: L אינה ריקה. }

פ1 ← (L, L) -פשימה (פ1, L)

פ2 ← (L, L) -פשימה (פ2, L)

כל פ1 ≠ פ2 פשימה

אם אחר-פשימה (L, פ1) = 1 אזי

(L, פ1) -פשימה

(L, פ2, 1) -פשימה

אחרת

פ1 ← (L, פ1) -פשימה

- א. כיצד תראה הרשימה הבאה לאחר סיום ריצת האלגוריתם? L: 1, 0, 1, 1, 0
- ב. מהי הפעולה אותה מבצע האלגוריתם?
- ג. אם ידוע שמספר האיברים ברשימה הוא N, מהי סיבוכיות האלגוריתם? נמק.



4.7 - היפוך רשימה

נקודת מוצא: רשימה מאותחלת.

מטרה: הפיכת סדר איברי הרשימה.

אלגוריתם: היפוך-רשימה (L)

$p \leftarrow (L, (L))$ *הפוך-רשימה (L) ל- (L)*

$p \leftarrow (L, p)$ *הפוך-רשימה (L) ל-p*

$p \neq (L)$ *הפוך-רשימה (L) ל-p*

$x \leftarrow (L, p)$ *הפוך-רשימה (L) ל-p*

(L, p) *הפוך-רשימה (L) ל-p*

$(L, (L))$ *הפוך-רשימה (L) ל- (L)*

הפוך את L

הערות

- שיטת ההיפוך שבתבנית היא סריקה של איברי הרשימה L מתחילתה ועד סופה, תוך העברת איבר-איבר לתחילת הרשימה. תחילה "יושאר" האיבר הראשון בתחילת הרשימה, לאחר מכן, יועבר האיבר השני לתחילתה ויושם לפני הראשון, אחר-כך יועבר השלישי ויושם בתחילת הרשימה, לפני השני, אחר-כך יועבר הרביעי להתחלה ויושם לפני השלישי, וכך הלאה. כלומר, ניתן לראות את אלגוריתם ההיפוך כסדרת הזזות מעגליות בתוך תת-רשימה שהולכת ומתארכת – תחילה אורכה 1, אחר-כך 2, אחר-כך 3, 4 וכך הלאה.
- תהליך ההיפוך הוא סדרה של העברות, שבהן מתהפך הסדר בין כל איבר מועבר לבין כל האיברים שהועברו לפניו. העברות אלו שונות באופיין מאלו שתוארו בתבנית הקודמת, תבנית 4.6, אשר בה נשמר הסדר היחסי בין האיברים המועברים.
- דרך אחרת לבצע היפוך היא ליצור רשימה חדשה, שבה יושם תחילה האיבר האחרון ב-L, ואחריו האיבר הלפני-אחרון ב-L, וכך הלאה. דרך זו היא אינטואיטיבית, אך מאד לא-יעילה, כיון שההתקדמות ב-L נעשית מהסוף להתחלה וכרוכה בסריקה חוזרת ונשנית של הרשימה L.
- יתכן שלפעמים יהיה צורך להפוך רק חלק מרשימה. אם החלק הוא החלק התחילי (למשל, היפוך הסדר של 10 האיברים הראשונים), אזי האלגוריתם יהיה דומה מאד לזה שבתבנית. אם החלק הוא חלק אחר ב-L, אזי שיטת ההיפוך תהיה דומה, אך חישוב וציון מקום ההכנסה של האיבר המועבר הבא יהיו שונים.

<pre> (*) פעולה המקבלת רשימה L והופכת את סדר איבריה כך שהאיבר האחרון יהיה בראש הרשימה ואילו האיבר הראשון יהיה בסופה. הנחה: הרשימה L מאותחלת. *) procedure list_reverse (var L : list_type); var pos, p : pos_type; x : list_info_type; begin pos := list_anchor (L); p := list_next (L, pos); p := list_next (L, p); { p מצביע על האיבר השני ב-L } while p <> list_end (L) do begin list_retrieve (L, p, x); list_delete (L, p); list_insert (L, pos, x); end; end; end; </pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"><i>Pascal</i></div>
---	--

<pre> /* פעולה המקבלת רשימה L והופכת את סדר איבריה כך שהאיבר האחרון יהיה בראש הרשימה ואילו האיבר הראשון יהיה בסופה. הנחה: הרשימה L מאותחלת. */ void list_reverse (list_type L) { pos_type pos, p; list_info_type x; pos = list_anchor (L); p = list_next (L, pos); p = list_next (L, p); // p מצביע על האיבר השני ב-L while (p != list_end (L)) { x = list_retrieve (L, p); list_delete (L, &p); list_insert (L, pos, x); } } </pre>	<div style="border: 1px solid black; padding: 2px; display: inline-block;"><i>C</i></div>
--	---

שאלות

שאלה 4.7.1

- א. פתח אלגוריתם שיקבל רשימה L וישרשר אותה לעצמה בצורה הפוכה, כך שהחלק הראשון של הרשימה המשורשרת יהיה הרשימה L המקורית ואילו החלק האחרון יהיה היפוכה של L .
- ב. ממש בסביבת העבודה את האלגוריתם שכתבת בסעיף הקודם.

■

שאלה 4.7.2

לפניך אלגוריתם רקורסיבי הפועל על רשימה של אותיות.

שנה-רשימה (L)

{ טענת כניסה : רשימה מאותחלת של אותיות.

טענת יציאה : _____ }

אם לא רשימה-פיקה? (L) אפי

אחרון-רשימה (L) $p \leftarrow$

אחרון-רשימה (L, p) $x \leftarrow$

רשימה-רשימה (L, p)

שנה-רשימה (L)

הכנס-רשימה (x, א. אחרון-רשימה (L, L))

- א. פתח אלגוריתם למימוש הפעולה : **אחרון-רשימה (L)** המקבל רשימה מאותחלת L ומחזיר את מקומו של האיבר האחרון ברשימה. אם הרשימה L ריקה יחזור המקום עוגן-רשימה.
- ב. מה מבצע האלגוריתם **שנה-רשימה (L)** ?
- ג. מהי סיבוכיות האלגוריתם? נמק!

■

שאלות סיכום

שאלה 1

לפניך פעולות המוגדרות על רשימה הממומשת על-ידי שרשרת חוליות :

<p>הפעולה מחזירה איבר שנבחר באקראי מתוך רשימה L. <u>הנחות</u> : L רשימה מאותחלת ולא ריקה.</p>	<p>בחר-אקראי (L)</p>
<p>הפעולה מקבלת רשימה L המכילה מספרים שלמים ואיבר x מטיפוס איברי הרשימה ומחזירה רשימה חדשה המכילה את כל המספרים מתוך L הקטנים מ- x. <u>הנחות</u> : L מאותחלת.</p>	<p>רשימת-איברים-קטנים (L, x)</p>
<p>הפעולה מקבלת רשימה L המכילה מספרים שלמים ואיבר x מטיפוס איברי הרשימה ומחזירה רשימה חדשה המכילה את כל המספרים מתוך L הגדולים מ- x. <u>הנחות</u> : L מאותחלת.</p>	<p>רשימת-איברים-גדולים (L, x)</p>
<p>פעולה המקבלת רשימה L ומחזירה את מספר האיברים בה. אם הרשימה ריקה תחזיר הפעולה 0. <u>הנחות</u> : L מאותחלת.</p>	<p>מספר-איברים (L)</p>
<p>פעולה המוחקת מרשימה L את k האיברים הקטנים ביותר. <u>הנחות</u> : L מאותחלת ומונה לפחות k איברים.</p>	<p>מחק-איברים-קטנים (L, k)</p>
<p>פעולה המוחקת מרשימה L את k האיברים הגדולים ביותר. <u>הנחות</u> : L מאותחלת ומונה לפחות k איברים.</p>	<p>מחק-איברים-גדולים (L, k)</p>

האלגוריתם הבא פועל על רשימה L, המכילה מספר אזוגי של מספרים שלמים, שכולם שונים זה מזה.

מי-אני? (L)

בחירה-אקראית (L) $x \leftarrow$

פשימת-איבריט-קטן (L, x) $L1 \leftarrow$

מספר-איבריט (L1) $k1 \leftarrow$

פשימת-איבריט-טוב (L, x) $L2 \leftarrow$

מספר-איבריט (L2) $k2 \leftarrow$

אם $k2 < k1$ אזי

מחק-איבריט-קטן (L1, k2+1)

החזר מי-אני? (L1)

אם $k2 > k1$ אזי

מחק-איבריט-טוב (L2, k1+1)

החזר מי-אני? (L2)

אם $k2 = k1$ אזי

החזר x

- א. מה יחזיר האלגוריתם עבור הרשימה הבאה: $L: 43, -3, 8, 9, 12, -9, 7, 3, 1$?
 ב. מה מטרת האלגוריתם?
 ג. ממש את הפעולה **מחק-איברים-גדולים** (L, k).
 ד. מהי סיבוכיות הפעולה שמימשת בסעיף ג? נמק.

■

שאלה 2

רשימה דו-כיוונית היא רשימה במימוש שרשת חוליות שלכל חוליה בה מצביע נוסף המצביע לחוליה הקודמת.

- א. אילו שינויים יש לבצע בממשק **רשימה** כדי להתאימו ל**רשימה דו-כיוונית**?
 ב. אילו שינויים יש לבצע במימוש הפעולות על **רשימה** כדי להתאימה ל**רשימה דו-כיוונית**?
 ג. כיצד תשתנה סיבוכיות הפעולות ב**רשימה דו-כיוונית**?
 ד. כתוב יחידת ספריה עבור **רשימה דו-כיוונית**.

■

שאלה 3

למכשיר טלפון סלולארי פונקציות מתקדמות הכוללות שמירת ספר טלפונים ממויין, רשימת שיחות יוצאות, רשימת שיחות נכנסות וכו'.
להלן חלק ממודול **טלפון-סלולארי** המטפל ברשימת הנמענים השמורה במכשיר :

רשימת-טלפונים (L)	פעולה המציגה רשימה ממוינת של מספרי הטלפון השמורים במכשיר. <u>הנחה</u> : הרשימה מאותחלת.
הוספת-נמען (L, person)	פעולה המקבלת נמען חדש person ומוסיפה אותו לרשימת הטלפונים השמורה במכשיר בצורה ממוינת לפי שם הנמען. <u>הנחות</u> : הרשימה מאותחלת. הנמען person מאותחל.
איתור-מספר (L, name)	פעולה המחזירה את פרטי הנמען ששמו name מרשימת הטלפונים השמורה במכשיר. אם הנמען לא קיים במכשיר יוחזרו פרטי הנמען הראשון שאחרי הנמען name במיון הא"ב. אם אין אחריו נמענים יוחזרו פרטי הנמען הראשון ברשימה. <u>הנחה</u> : הרשימה מאותחלת ולא ריקה.
מחיקת-נמען (L, person)	פעולה המקבלת נמען person ברשימה L ומוחקת אותו מהספר. <u>הנחות</u> : הרשימה מאותחלת. P מקום של נמען ברשימה.
שיחה-מזוהה (L, num)	פעולה המקבלת מספר טלפון ומחזירה את שם הנמען שמתאים למספר זה. אם הנמען אינו קיים במאגר יוחזר 'לא-מזוהה'. <u>הנחות</u> : הרשימה מאותחלת.

פתח אלגוריתם למימוש כל אחת מהפעולות המופיעות בממשק **טלפון-סלולארי**.

■

שאלה 4

פתח אלגוריתם בשם **שלב-תת-רשימה** ($L1, p, L2$) המקבל כפרמטר שתי רשימות מאותחלות L1 ו-L2 ומקום p ב-L1, ומכניס את כל איברי רשימה L2 לתוך רשימה L1 אחרי המקום p. לדוגמה:

p
↓

L1: 2, 5, 1, 9, 4, 8	אם הרשימות לפני הפעולה היו:
L2: 3, 7, 6	
L1: 2, 5, 1, 3, 7, 6, 9, 4, 8	אחרי השילוב תהיה:

■

שאלה 5

נתונה רשימה המורכבת מתת-סדרות לא ריקות של מספרים, כך שכל תת-סדרה מסתיימת באפס. תת-הסדרות אינן ממוינות. פתח אלגוריתם שימחק את תת-הסדרה בעלת הסכום הנמוך ביותר. לדוגמה: עבור הרשימה הבאה: $L: 2, 7, -1, 0, -5, 4, 1, 6, 0, 7, 9, 0, 8, 3, -2, 5, 4, 0$ תימחק תת-הסדרה $0, 6, 1, 4, -5$ שסכומה 6 הוא הסכום הנמוך ביותר.

■

שאלה 6

ב"מכרז השבוע" מציעים המתמודדים הצעות מחיר למוצר השבועי. כל מתמודד נרשם ע"י כתיבת שמו, מספר הטלפון שלו, גובה ההצעה שהציע, ושעת קבלת ההצעה בחברה. קביעת הזוכים במכרז תתבצע באופן הבא: ההצעות יסודרו מהצעת המחיר הגבוהה ביותר לנמוכה ביותר. מספר הזוכים ייקבע בהתאם לכמות הפריטים במלאי. במקרה ששני מתמודדים נתנו הצעה זהה, יזכה המתמודד שהצעתו התקבלה ראשונה.

א. פתח אלגוריתם לטיפול בתוצאות המכרז של שבוע מסוים. האלגוריתם יקבל את נתוני ההצעות, לפי שעת הקבלה שלהן, ויודיע מיהם הזוכים.

ב. ממש את האלגוריתם בסביבת העבודה.

ג. מהי סיבוכיות האלגוריתם שכתבת?

■

שאלה 7

נתונות שתי רשימות $L1$ ו- $L2$. פתח אלגוריתם שיקבל כפרמטר את הרשימות ויחזיר 'אמת' אם $L2$ מופיעה במהופך בתוך $L1$, ו-'שקר' אחרת.

■

