

מבוא למדעי המחשב

מטלה 4

נושא: מערכים

מתרגל אחראי: אביב שוקרון

הוראות כלליות

המטלה כוללת 3 קבצים:

mainTrain.c – קובץ ה-main שלכם, הוא מריץ בדיקות על הפונקציות שתממשו.
Utils.h – קובץ הגדרות הפונקציות של התכנית.
Utils.c – קובץ בו תממשו את הפונקציות של התכנית.

במטלה זו תתבקשו לממש 4 פונקציות.

יש להוריד את המטלה ממערכת ההגשות האוטומטית ולהשתמש בקובץ ה-H המקורי על מנת שלא יתגלו שגיאות קומפילציה. למשל, שלא רשמתם שמות של פונקציות בצורה נכונה.

שימו לב!

בכל שאלה אתם רשאים להשתמש בקוד שמימשתם מסעיפים קודמים. למשל אם הפונקציה משאלה מספר 2 יכולה להשתמש בפונקציה שמימשתם בשאלה 1 אתם רשאים לקרוא לפונקציה מ-1 בפונקציה של 2.

לאחר שתממשו, תוכלו לקמפל ולהריץ את התכנית עם ה-main המוכן או לשנות את ה-main כראות עיניכם ולבצע עוד בדיקות משלכם.

חשוב!

יש לוודא הרצה של התכנית בסביבה שלכם לפני הגשה למערכת האוטומטית. בצורה זו תוכלו ללמוד ולהתפתח בצורה הטובה ביותר.

בסיום המטלה הנכם נדרשים להגיש **רק קובץ אחד – Utils.c** ובו המימושים של השאלות.

לא לשכוח לרשום הערות לאורך הקוד (כל 2-3 שורות).

כמו כן הנכם מתבקשים לרשום הערה ארוכה בתחילת הקובץ Utils.c הכוללת שם, ת.ז ותאריך. כמו כן הנכם מתבקשים לרשום לפני כל פונקציה הערה ובה מה הפונקציה עושה.

1. `int RemoveDuplicates(int arr[], int size);`
הפונקציה מקבלת כפרמטר מערך למספרים שלמים וגודלו. הפונקציה תוריד את הכפילויות מהמערך תוך צמצום הרווחים ותחזיר את גודלו הנוכחי לאחר הורדת הכפילויות.

דוגמא:

`arr = 1, 2, 1, 4, 4, 3, 5, 2, 1`
`size = 9`

הפונקציה תשנה את `arr` ל-

`arr = 1, 2, 4, 3, 5, x, x, x, x`
הסבר – סדר המספרים חייב להישמר לפי סדר הופעתם במקור, כך למשל 4 הופיע לפני 3 ולכן גם בתוצאה יהיה כך. מה שמסומן כ-x אלה המספרים שהיו כפולים. לא חשוב שם הסדר, העיקר שכל הכפולים יעבור לסוף.
כמו כן הפונקציה תחזיר 5, מאחר ויש סה"כ 5 ערכים שונים במערך.

• אין להשתמש בשאלה זו במערך עזר!

2. `void InsertionSort(int arr[], int size);`
הפונקציה תקבל כפרמטר מערך למספרים שלמים ואת גודלו. הפונקציה תבצע מיון למערך בעזרת אלגוריתם "מיון הכנסה".

"מיון הכנסה" הוא אלגוריתם פשוט למיון מערך, הוא עובר מספר פעמים על המערך ובכל מעבר משנה את סדר האיברים, עד שמתקבל המצב הרצוי שבו סדר האיברים עולה. לשם כך הוא נעזר בתא זיכרון בודד ובשני מונים. בכל מעבר על המערך, נלקח איבר אחד ומוכנס למקומו הנכון בתוך המערך הממוין שנבנה, במהלך המיון, בחלק השמאלי של המערך. לאחר השלב הראשון, כולל האזור הממוין במערך את שני האיברים הראשונים. לאחר השלב השני, כולל האזור הממוין המערך את שלושת האיברים הראשונים וכך הלאה...
הסבר נוסף על "מיון הכנסה" ניתן לראות כאן –

https://he.wikipedia.org/wiki/%D7%9E%D7%99%D7%95%D7%9F_%D7%94%D7%9B%D7%A0%D7%A1%D7%94

3. `void FlipMatrix(int mat[MATRIX_SIZE][MATRIX_SIZE], int option);`
הפונקציה מקבלת מטריצה (מערך דו מימדי של מספרים שלמים) ומספר המהווה אופציה לבחירה שעל פיה תבצע סיבוב או היפוך של המטריצה.
הערה – מספר השורות והעמודות במטריצה זהה.
אם האופציה היא 1 – הפונקציה תסובב את המטריצה ב-90 מעלות ימינה.
אם האופציה היא 2 – הפונקציה תסובב את המטריצה ב-90 מעלות שמאלה.
אם האופציה היא 3 – הפונקציה תהפוך את המטריצה מלמעלה למטה (תמונת מראה).
אם האופציה היא 4 – הפונקציה תהפוך את המטריצה משמאל לימין (תמונת מראה).

שימו לב! במערכת הבדיקות הגודל `MATRIX_SIZE` יכול להיות גדול או קטן מ-4, לכן כתבו אלגוריתם גנרי שיעבור עבור כל גודל ולא ספציפי עבור מטריצה בגודל 4x4.

דוגמא:

המטריצה:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Option = 1:

13	9	5	1
14	10	6	2
15	11	7	3
16	12	8	4

Option = 2:

4	8	12	16
3	7	11	15
2	6	10	14
1	5	9	13

Option = 3:

13	14	15	16
9	10	11	12
5	6	7	8
1	2	3	4

Option = 4:

4	3	2	1
8	7	6	5
12	11	10	9
16	15	14	13

4. `int CountSemiSimilar(char a[][M], char b[][M], int n);`
 הפונקציה מקבלת 2 מערכים של מחרוזות ומספר שלם וחיובי n . המערך הראשון מכיל n מחרוזות a והמערך השני מכיל n מחרוזות b . הפונקציה תחזיר מספר שלם חיובי המציין את כמות הזוגות של (a_i, b_i) אשר מהוות זוג מחרוזות "דומות למחצה".

שתי מחרוזות a ו- b בעלי אורכים זהים נקראות "דומות למחצה" אם המחרוזת a זהה למחרוזת b לאחר **החלפת מיקומי תווים** במחרוזת b .

לדוגמא:

עבור: $a = abcdefg$ ו- $b = cdfegba$.

קל לראות כי אם נחליף את מיקומי התווים במחרוזת b אנו נקבל את המחרוזת a .

מגבלות:

- המספר n גדול או שווה ל-1 וקטן או שווה מ- M .
- אורך כל מחרוזת גדול מ-0 אך לא גדול יותר מ-10,000.
- המחרוזות מכילות אותיות קטנות באנגלית בלבד.
- יש להניח כי n תקין.
- ייתכן כי $|a_i| \neq |b_i|$ ובמקרה כזה המחרוזות **אינן דומות למחצה**.

דוגמא:

עבור $n = 3$ ועבור המערכים הבאים:

a :

a	b	c	d	e	f	g	\0
g	f	e	g	\0			
a	b	c	d	\0			

b :

c	d	f	e	g	b	a	\0
g	e	f	f	\0			
a	b	c	d	e	\0		

הפונקציה תחזיר את הערך 1.

- זוג המחרוזות הראשון תקין (ראה דוגמא קודמת)
- בזוג השני, אין כמות זהה של אותיות g ואותיות f ולכן לא מהוות מחרוזות דומות למחצה
- בזוג המחרוזות השלישית, המחרוזות לא באותו אורך, ולכן לא מהוות מחרוזות דומות למחצה

סה"כ מוחזר 1.