

תרגיל בית מספר 5

מצביעים והקצאות דינאמיות

הוראות כלליות

המטלה כוללת 3 קבצים:
mainTrain.c – קובץ ה-main שלכם, הוא מריץ בדיקות על הפונקציות שתממשו.
Util.h – קובץ הגדרות עבור הפונקציות.
Util.c – קובץ מימושים עבור הפונקציות.

במטלה זו תתבקשו לממש 6 פונקציות. המימושים צריכים להיות בתוך ההגדרות שמוכנות לכם בקובץ Util.c.
לאחר שתממשו, תוכלו לקמפל ולהריץ את התכנית עם ה-main המוכן או לשנות את ה-main כראות עיניכם ולבצע עוד בדיקות משלכם.

בסיום המטלה הנכם נדרשים להגיש **קובץ אחד בלבד – Util.c** ובו המימושים של השאלות.

לא לשכוח לרשום הערות לאורך הקוד (כל 2-3 שורות).
כמו כן הנכם מתבקשים לרשום הערה ארוכה בתחילת התכנית הכוללת שם, תז, תאריך ומה התכנית עושה.

שימו לב: בשאלות 1-6 אסור להשתמש באופרטור [].
חובה לעבור על איברי מערכים רק באמצעות פוינטרים.
תרגילים 1-4 ללא הקצאות דינאמיות

תרגיל 1

כתוב את הפונקציה void ToString(char *str, int number).
הפונקציה מקבלת מחרוזת ריקה (בגודל MAX_SIZE) ומספר **חיובי** כלשהו (גדול מ-0).
הפונקציה תמיר את המספר number לייצוג באמצעות מחרוזת, ותבצע השמה של מחרוזת זו ל-str.
אפשר להניח שב str יש מספיק מקום להכיל את הייצוג המחרוזתי של number.
לדוגמה, אם קיבלתם מחרוזת ריקה ואת המספר 1234, אזי הפונקציה תבצע את ההשמה הבאה ל-str:
"1234" (ז"א 5 תווים – '0','1','2','3','4').
(אין להשתמש בפונקציה ב- itoa המבצעת המרה של מספר int למחרוזת)

תרגיל 2

כתבו פונקציה `void Mystrcat(char* s, char* t)` המעתיקה את המחרוזת `t` לסופה של המחרוזת `s` (ניתן להניח שב-`s` יש מספיק מקום להכיל את שתי המחרוזות). (אסור להשתמש בפונקציה `!strcat`)
דוגמא:

```
char source[MAX_SIZE] = "I Love ";  
Mystrcat(source, "you")
```

כעת המחרוזת `source` תכיל:

"I Love you"

תרגיל 3

כתוב את הפונקציה `int SumStr(char *str)` הפונקציה תחזיר את סכום המספרים בתוך המחרוזת. לדוגמה, הקלט `ax3b5mt11f` והפלט יהיה 19 ($19=3+5+11$).
הערה: ניתן להשתמש בפונקציה `atoi(const char *str)` המקבלת מחרוזת המכילה מספר ומחזירה את המספר השלם למידע נוסף:

https://www.tutorialspoint.com/c_standard_library/c_function_atoi.htm

תרגיל 4

הגדרה: חציון של סדרת מספרים (סופית) הוא מספר הגדול או שווה לבדיק מחצית מאברי הסדרה, והקטן או שווה לבדיק מחצית מאברי הסדרה.
לדוגמה: סדרת המספרים 1,2,22,7,19,8 - סדרה בת 6 מספרים - המספר 7 הינו חציון, מפני שהוא גדול או שווה ל-1,2,7, וקטן או שווה ל-8,19,22.
(גם 8 הינו חציון, מפני שהוא גדול או שווה ל-1,2,7 וקטן או שווה ל-8,19,22)

כתוב את הפונקציה `int Median(int* numbers, int size)` המקבלת פוינטר למערך של מספרים ואת גודל המערך.
הפונקציה תמצא את איבר החציון ותחזיר אותו.
במידה ויש 2 חציונים (מערך בעל מספר איברים זוגי) הפונקציה תחזיר את החציון הקטן יותר.
במערך בעל מספר אי זוגי של מספרים, יש חציון יחיד.

תרגיל 5

כתוב את הפונקציה: `int* CreateRange(int start, int end, int jump, int *sizeRange)`.
הפונקציה מקבלת 3 מספרים שלמים המהווים מספר מינימלי (`start`) ומספר מקסימלי (`end`) ומספר שלם המהווה הפרש סדרה (`jump`).
הפונקציה תחזיר מצביע לתחילת מערך של מספרים שלמים (שיווצר באמצעות הקצאה דינמית - `malloc`) אשר האיבר הראשון בו הוא `start`, ושאר המספרים הינם בקפיצות של `jump` עד המספר הקרוב ביותר ל-`end`.
כמו כן, הפונקציה תאכסן במשתנה `sizeRange` את כמות האיברים המאוחסנים במערך.
יש להניח כי `start <= end`. במקרה ולא ניתן לבצע הקצאה דינמית יש להחזיר `NULL`.

לדוגמא:

1	4	7	10
---	---	---	----

בהינתן `start = 1, end = 11, jump = 3` אזי הפונקציה תחזיר את המערך
ובמשתנה `sizeRange` יהיה את הערך 4

תרגיל 6

כתוב את הפונקציה `char* PairArrays(char a[], char b[])` ומחרוזת `a`, ומחרוזת `b` ומחזירה מחרוזת חדשה (באמצעות הקצאה דינמית - `calloc`) אשר מהווה שילוב בין המחרוזות באופן הבא:
התו הראשון במחרוזת החדשה יהיה התו הראשון במחרוזת `a` והתו השני במחרוזת החדשה יהיה התו הראשון של מחרוזת `b`, התו השלישי במחרוזת החדשה הינו האיבר השני במחרוזת `a`, וחוזר חלילה.
אם המחרוזות אינן בגודל זהה, יש לשרשר את מה שנותר מהמחרוזת הגדולה יותר לסוף המחרוזת המשולבת.
במקרה ולא ניתן לבצע הקצאה דינמית יש להחזיר `NULL`.

מותר השימוש ב- strlen.

לדוגמא: עבור המחרוזות הבאות:

a:

a	b	c	\0
---	---	---	----

b:

v	w	x	y	z	\0
---	---	---	---	---	----

תוחזר המחרוזת:

a	v	b	w	c	x	y	z	\0
---	---	---	---	---	---	---	---	----