מבוא למדעי המחשב, סמסטר חורף 2019-2020

תרגיל בית 2

מועד אחרון להגשה: יום ד', 27/11 עד שעה 23:55

המתרגל האחראי על תרגיל זה: **גסוב מזאוי**

- **משרד**: טאוב 219
- **email**: Gasob.mazzawi@hotmail.co.il
  - **על נושא המייל להתחיל במספר הקורס (234114/234117) והתרגיל ולהמשיך בנושא השאלה שתופיע בגוף המייל, לדוגמא**
  234114 – HW 2 - Question 1 – using functions.
  - **טרם הפנייה, בדקו בדף ה-FAQ של התרגיל אם השאלה שלכם כבר נענתה.**

הנחיות:

- הגשה ב**בודדים**. עליכם לכתוב את הפתרונות לבד ולהגיש ביחידים.
- קראו את השאלות בעיון לפני שתתחילו בפתרונן.
- הקפידו לתעד את הקוד שלכם בהערות באנגלית.
- מלבד מילואים, לא יתקבלו תרגילים אחרי מועד הגשה. הגשה באיחור לאחר מועד הגשה נחשבת כאי-הגשה.
- כל יום מילואים = יום דחייה. על מנת לקבל את הדחייה, עליכם לשלוח באי-מייל למתרגל האחראי עותק של האישור המראה שהייתם במילואים (טופס 3010). אם האישור יגיע אליכם בתאריך מאוחר, יש להודיע על כך למתרגל האחראי.
- **לא ניתן לערער על תוצאות הבדיקה האוטומטית.**
- **שימו לב! הבדיקה הינה אוטומטית, ולכן הקפידו להדפיס בדיוק בפורמט שהתבקשתם ובידקו עם אתר הבדיקה ועם DiffMerge את הפלט שלכם מול הפלט של הדוגמאות שקיבלתם.**
  - השתמשו ב-redirection כדי להפנות את הפלט לקובץ טקסט.
  - השתמשו באתר הבדיקה העצמית לבדיקה וקבלת פלט צפוי.
  - השוו עם diffmerge (ראו דגשים בסוף הגיליון).
  - אין להדפיס רווחים ותווים שלא התבקשתם להדפיס.
- בתרגיל זה מותר להשתמש בפונקציות מהספרייה stdio.h למעט במקרים בהם נאמר אחרת. החומר הנדרש לתרגיל זה שייך להרצאות 1-5 ולתרגולים 1-5. אין להשתמש בחומר שאינו מופיע במצגות אלה (מותר להשתמש בלולאות ובמערכים).
- ההגשה הינה אלקטרונית ו**בבודדים** דרך אתר הקורס. קובץ ההגשה יהיה מסוג **zip** (ולא אף פורמט אחר) ויכיל בתוכו את הקבצים הבאים בלבד, ללא כל תיקיות:
  - קובץ **students.txt** עם שמך **באנגלית**, מספר תעודת הזהות וכתובת האי-מייל שלך.
  - קובץ פתרון **hw2q1.c** עבור שאלה 1.
  - קובץ פתרון **hw2q2.c** עבור שאלה 2.
  - קובץ פתרון **hw2q3.c** עבור שאלה 3.
- **חובה לשמור את אישור ההגשה שמקבלים מהמערכת לאחר שמגישים, עד לסיום הקורס.**
- יש להקפיד להגיש את כל הקבצים בדיוק עם השמות שמופיעים לעיל. הגשה שלא תעמוד בתנאי זה **לא תתקבל ע"י המערכת!** אם המערכת לא מקבלת את התרגיל שלכם, חפשו את הפתרון לבעיה באתר הקורס תחת הכפתור FAQ.

עם כל תרגיל פורסמו 4 טסטים (קלט-פלט צפוי) כדי שתוכלו לבדוק את עצמכם, מקרי ההרצה המפורטים שלעיל הינם טסטים 1-4. שימו לב, הציון יתבסס על מקרים נוספים שאנו נבדוק ולכן חשוב שתבדקו את התוכנית על מקרים נוספים ולא רק על הטסטים שפרסמנו.

שימו לב:

באתר הקורס פורסמו 3 קבצי שלד לשלושת השאלות בתרגיל תחת השמות :

hw2q1.c
hw2q2.c
hw2q3.c

בקבצים אלו קיימות כבר פונקציות הדפסה (שהשמות שלהם מתחילים ב print) שבאמצעותם תדפיס התוכנית שלכם לפלט.
אתם יוצרים את הפרויקטים שלכם לכל שאלה (לדוגמה פרויקט בשם **hw2q1** כפי שנלמד בעבר ואיך שעשיתם בתרגילים קודמים) , אך תחליפו את קובץ ה C שיוצר בפרויקט בקובץ המתאים לשאלה זו שפורסם באתר לדוגמה (**c.hw2q1** עבור פרויקט של שאלה 1) ולשם תמשיכו להוסיף את הקוד שלכם לפתרון השאלות אם זה קוד לתוך ה main או מימוש פונקציות נוספות מחוץ ל main.
ואלו הם שלושת הקבצים שגם תגישו יחד עם קובץ **students.txt**.
בכל שאלה בתרגיל אתם יכולים להוסיף פונקציות עזר כרצונכם.

בדיקה ידנית:

בנוסף לבדיקה האוטומטית, התרגיל ייבדק גם בבדיקה ידנית. הבדיקה תתמקד בנושאים הבאים:

- רוחב שורה - רוחב כל שורה (**כולל הערות והזחות**) לא יעלה על 150 תווים. ניתן לסמן אורך של שורה בקודבלוקס.
- קבועים בקוד:
  - יש להגדיר בעזרת #define כל קבוע משמעותי. שמות קבועים צריכים להיות באותיות גדולות בלבד, אם השם מכיל יותר ממילה אחת מילים יופרדו בעזרת מקף תחתון (למשל STUDENTS_NUM).
  - אין להשתמש בערכי ASCII ישירות. יש להשתמש בייצוג של התווים (למשל 'a').
- הזחות:
  - שיטת הזחה מקובלת – הזחת קוד בכל בלוק, למשל:

```
int main()
{
	// your code here
	while (…)
	{
		// your code here
	}
}
```

- אין להשתמש במשתנים גלובאליים או סטטיים.
- שמות משתנים/קבועים/פונקציות צריכים להיות אינפורמטיביים, להעיד על מטרתם.
- בהירות הקוד ותיעוד:
  - יש לתעד את הקוד באמצעות הערות באנגלית בלבד. במידה ויש כמה שורות קוד שניתן להסביר בקצרה מה המטרה שלהן, יש לשים הערה בהתחלה ואין צורך לתעד כל שורה.
  - יש לתעד פונקציות – לפני הפונקציה להוסיף הערה שמסבירה בקצרה מה הפונקציה עושה ומה המשמעות של הפרמטרים שלה.
  - התיעוד צריך להיות אינפורמטיבי, כלומר יש להסביר מה המטרה של שורות הקוד ולא לכתוב את הקוד במילים.
- שכפול קוד שלא לצורך (למשל ריבוי קוד זהה במספר מקרי if-else שונים).
- אי-עמידה באחת מדרישות התרגיל (שימוש בחומר שהיה אסור בתרגיל וכו').

באופן כללי – הקפידו על כתיבת קוד מסודר ומובן ככל שניתן תוך יישום העקרונות שנלמדו בכיתה.
מותר לכם לממש פונקציות עזר משלכם ולהשתמש בהן.

בונוס (10 נקודות לציון התרגיל):

אם עומדים בדרישה הנ"ל:

- אורך כל פונקציה לא יעלה על 16 שורות קוד. הגבלה זו תקפה לכל הפונקציות, כולל main.
רק שורות ריקות, שורות עם סוגר מסולסל בלבד, ושורות עם הערות בלבד לא נספרות.
אסור לכתוב כמה פקודות שונות משמעותית באותה השורה, למשל כותרת תנאי\לולאה צריכה להיות בשורה נפרדת מגוף התנאי\לולאה.

שאלה 1: אופרטורים לוגיים ואריתמטיים

Lazy evaluation - תזכורת:

כפי שנלמד בהרצאה , חישוב אופרטורים לוגיים מתבצע בשיטת "חישוב עצל", כלומר:

* האסוציאטיביות של אופרטורים לוגיים הינה משמאל לימין.

* אופרטורים של השוואה קודמים לאופרטורים לוגיים.

* חישוב ערך של ביטוי בו מופיעים אופרטורים לוגיים נעצר ברגע שערך הביטוי אנו ניתן לשינוי.

כמו כן, חישוב ביטוי לוגי **מניב ערך החזרה** "false" / "true" .

נתבונן בקטע הקוד הבא לצורך המחשה:

```
bool res = (expression1) && printf("A");
```

לפי שיטת החישוב העצל, **רק** במידה והביטוי הראשון מניב "true" פקודת ההדפסה תבוצע. אחרת הפקודה לא תבוצע.

הערה: ע"מ לבצע חישוב של ביטוי לוגי יש לקלוט את תוצאתו. בתרגיל זה אפשר להגדיר משתנה למטרה זו (בד"כ לא נהוג להגדיר משתנה שלא מבצעים שימוש בערכו, אולם בתרגיל זה אנו מאפשרים זאת לצורכי לימוד).

וכעת לתרגיל עצמו...

במהלך מסעה לכבוש את כיסא הברזל, נאלצת המלכה תאליסי להילחם באויבים רבים.

לשם כך, עומדים לרשותה 3 דרקונים: dragonA , dragonB, dragonC.

על אף עליונותה, מעוניינת תאליסי לנהל קרב הוגן. לכן, תאליסי שולחת לכל היותר דרקון אחד לכל קרב. בחירת הדרקון נעשית על בסיס שמו של האויב באופן הבא:

1. אם סכום הערכים האסקיים של האותיות המרכיבות את שמו של האויב לא מתחלק ב-5 או אם שמו של האויב עולה על 4 תווים - dragonA יישלח לקרב.

2. אם הערכים האסקיים של האותיות המרכיבות את שמו של האויב מהווה סדרה מונוטונית עולה ממש – dragonB יישלח לקרב.

3. אם שמו של האויב קצר מ-6 תווים וגם אינו מכיל את האות 's' – dragonC יישלח לקרב.

כמו כן, במידה ובאפשרותה של תאליסי לשלוח לקרב יותר מדרקון אחד, תבחר תאליסי לשלוח את הדרקון הגדול ביותר לקסיקוגרפית (dragonA < dragonB < dragonC).

במידה ואף תנאי אינו מתקיים, תאליסי לא תשלח אף דרקון לקרב.

עליכם לכתוב תוכנית הקולטת רצף תווים המסתיים בתו " ! " ומהווה את שם האויב הקורא תיגר על המלכה. על התוכנית להדפיס למסך את בחירתה של תאליסי לפי ההנחיות בשאלה.

דגשים למימוש:

- ניתן להניח כי שמו של האויב יורכב מאותיות קטנות בשפה האנגלית בלבד (אין צורך לבדוק תקינות קלט).
- שם האויב מסתיים בתו " ! " (תו זה אינו חלק משם האויב).

בונוס: (10 נקודות):

על פתרון ללא שימוש במשפטי תנאי כגון: if...else , switch, a ? b: c

דוגמאות הרצה:

```
Please enter enemy name:
lannister!

Khalisi sent dragonA into the battle!!!
```

```
Please enter enemy name:
adirxyz!

Khalisi sent dragonB into the battle!!!
```

```
Please enter enemy name:
dindin!

Khalisi sent dragonA into the battle!!!
```

```
Please enter enemy name:
sba!

NO Dragon has been sent into the battle!!!
```

שאלה 2: מערכים חד מימדיים

לאחר קרב עקוב מדם, החליט יועצה ויד ימינה של חאליסי, טיריון, כי ברצונו לייעל את הצבא.

לפיכך, החליט טיריון כי מעתה ואילך, יחולק הצבא ל-10 מחלקות בדיוק.

על מנת להקל על אופן החלוקה ולמנוע סכסוך פנימי בין הלוחמים, הוחלט לתת לכל לוחם מספר אישי בן 3 ספרות. חלוקת הלוחמים תעשה עפ"י ספרת האחדות של סכום הספרות המרכיבות את מספרו האישי של כל לוחם.

כלומר, לוחם מספר 230 ישובץ למחלקה מספר 5. לוחם מספר 940 ישובץ למחלקה מספר 3.

כמו כן, ליועץ 4 דרישות להצגת נתונים על אודות הצבא הממוספרות 1-4 לפי הסדר הבא:

1. גודל הצבא – סך כל הלוחמים בכל המחלקות יחדיו.

2. מספר המחלקה הגדולה ביותר וגודלה.

(במקרה של שוויון המחלקה בעלת האינדקס הגדול ביותר מנצחת)

3. מספר המחלקות בהן יש לפחות לוחם אחד. בנוסף, גודלה הממוצע של מחלקה לא ריקה עם דיוק של 2 ספרות אחר הנקודה.

4. הצגת כל המחלקות בפורמט הבא: בשורה הראשונה יש להדפיס את מספר המחלקה ומתחת לכל מחלקה תוצג עמודה של כוכביות (' * ') כמספר הלוחמים במחלקה.

(דוג' לפלט מצורפת בדוגמאות ההרצה).

כתבו קוד הקולט מספר לא מוגבל של לוחמים ומסתיים במספר הדרישה המבוקשת.

על התוכנית להדפיס למסך את הנתון התואם את דרישתו של היועץ לפי מספר הדרישה.

דגשים למימוש:

- **ניתן להניח תקינות הקלט (אין צורך בבדיקת תקינות).**
- **מספרי הלוחמים יוכנסו בנפרד האחד מן השני. לדוג': עבור שני לוחמים המזוהים ע"י 230 ו-940, הקלט יהיה 230 940 (ולא כרצף 940230).**
- **מספר לוחם הינו מספר חיובי גדול ממש מ-99.**

דוגמאות הרצה:

```
Enter warriors id and demand at the end:

100 200 201 900 948 124 300 323 4
Show army!

0123456789
 ***    ***
 * *
```

```
Enter warriors id and demand at the end:

125 100 978 154 155 347 3
There are 4 active platoons

The avg size of a platoon is: 1.50
```

```
Enter warriors id and demand at the end:

100 200 101 201 102 300 206 2
The biggest platoon index is: 3 and its size is: 3
```

```
Enter warriors id and demand at the end:

100 656 978 142 1
The army size is: 4
```

שאלה 3: מימוש מחשבון מטריצות

יואש לקח הסמסטר את הקורס אלגברה ליניארית, בו הוא נדרש לבצע פעולות כפל וחיבור מטריצות.

יואש התעצל לבצע את החישובים, ולכן החליט לכתוב תכנית שתממש מחשבון מטריצות.

עזרו ליואש לכתוב תכנית זו.

על התכנית לקבל מהמשתמש גודל מטריצות מבוקש n, שתי מטריצות של מספרים שלמים בגודל $n \times n$ ופעולת חשבון (חיבור או כפל) ולהדפיס את מטריצת התוצאה.

להזכירכם, חיבור מטריצות מתבצע באופן הבא:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} + \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} = \begin{pmatrix} a_{1,1}+b_{1,1} & a_{1,2}+b_{1,2} & a_{1,3}+b_{1,3} \\ a_{2,1}+b_{2,1} & a_{2,2}+b_{2,2} & a_{2,3}+b_{2,3} \\ a_{3,1}+b_{3,1} & a_{3,2}+b_{3,2} & a_{3,3}+b_{3,3} \end{pmatrix}$$

כפל מטריצות מתבצע באופן הבא:

$$\begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{pmatrix} * \begin{pmatrix} b_{1,1} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{2,2} & b_{2,3} \\ b_{3,1} & b_{3,2} & b_{3,3} \end{pmatrix} =$$

$$\begin{pmatrix} a_{1,1}*b_{1,1}+a_{1,2}*b_{2,1}+a_{1,3}*b_{3,1} & a_{1,1}*b_{1,2}+a_{1,2}*b_{2,2}+a_{1,3}*b_{3,2} & a_{1,1}*b_{1,3}+a_{1,2}*b_{2,3}+a_{1,3}*b_{3,3} \\ a_{2,1}*b_{1,1}+a_{2,2}*b_{2,1}+a_{2,3}*b_{3,1} & a_{2,1}*b_{1,2}+a_{2,2}*b_{2,2}+a_{2,3}*b_{3,2} & a_{2,1}*b_{1,3}+a_{2,2}*b_{2,3}+a_{2,3}*b_{3,3} \\ a_{3,1}*b_{1,1}+a_{3,2}*b_{2,1}+a_{3,3}*b_{3,1} & a_{3,1}*b_{1,2}+a_{3,2}*b_{2,2}+a_{3,3}*b_{3,2} & a_{3,1}*b_{1,3}+a_{3,2}*b_{2,3}+a_{3,3}*b_{3,3} \end{pmatrix}$$

ראשית המשתמש יתבקש להכניס את גודל המטריצה המבוקש. ניתן להניח כי מוכנס מספר שלם חיובי ממש, בטווח הגודל של int.

לאחר מכן המשתמש יתבקש להכניס את המטריצה הראשונה בגודל $n \times n$ (סדר הכנסת האיברים הוא משמאל לימין שורה אחר שורה). אם אחד האיברים שהכניס המשתמש אינו מספר שלם יש להדפיס "Invalid input.\n" ולסיים את התכנית. בשלב הבא המשתמש יתבקש להכניס את המטריצה השנייה בגודל $n \times n$ (סדר הכנסת האיברים כנ"ל). גם כאן, אם אחד האיברים שהכניס המשתמש אינו מספר שלם יש להדפיס "Invalid input.\n" ולסיים את התכנית. יש להניח כי הוכנסו מספרים בטווח הגודל של int.

לבסוף יתבקש המשתמש להכניס את הפעולה הרצויה '*' – כפל או '+' – חיבור. אם המשתמש הכניס תו שאינו '*' או '+', יש להדפיס "Invalid input.\n" ולסיים את התכנית.

שימו לב: ניתן להניח שגודל המטריצה שיתקבל כקלט קטן מ 25X25.

להלן דוגמאות הרצה. המטריצה האחרונה היא תוצאת החישוב של התכנית:

```
Please enter the matrix size:
3
Please enter the first matrix:
7 8 9
1 2 3
4 7 8
Please enter the second matrix:
45 1 23
8 5 5
7 4 1
Please enter the required operation:
*
442 83 210
82 23 36
292 71 135
```

```
Please enter the matrix size:
4
Please enter the first matrix:
7 8 9 10
1 2 3 4
-4 -7 8 5
5 17 29 8
Please enter the second matrix:
45 1 -29 2
8 5 5 5
7 4 1 33
10 10 80 0
Please enter the required operation:
+
52 9 -20 12
9 7 8 9
3 -3 9 38
15 27 109 8
```

```
Please enter the matrix size:
3
Please enter the first matrix:
7 8 9
1 2 3
4 7 8
Please enter the second matrix:
45 1 23
8 5 5
7 4 1
Please enter the required operation:
y
Invalid input.
```

דגשים נוספים :

יש להיעזר באתר הבדיקה האוטומטית http://csm.cs.technion.ac.il/~cs234114/ על-מנת לבדוק את הקוד שלכם. האתר מאפשר לכם לשלוח את הקוד שלכם לשאלה מסוימת (קובץ c.) ולבדוק האם הוא עובר בדיקות מסוימות בריצה של הבודק האוטומטי. התוצאה לכל אחת מהבדיקות יכולה להיות אחת משלוש:

א. "עבר" – הבדיקה עברה בהצלחה!.
ב. "נכשל" – הפלט עבור הבדיקה לא יצא זהה. במקרה כזה יש להפעיל את התוכנית באמצעות redirection כפי שנלמד בתרגול ובתרגיל בית 0 ולמצוא באמצעות diffmerge את ההבדלים (את הקלט והפלט המצופה לכל הבדיקות תוכלו למצוא באתר הקורס).
ג. "נתקע" – התכנית נתקעה בלולאה אינסופית או שהיא ממתינה לקלט (יש לחכות 30 שניות עד לקבלת התשובה).

במידה ותהיה בקוד שלכם שגיאת קומפילציה כל הבדיקות יקבלו תוצאת "נכשל" והשגיאה עצמה תהיה רשומה במפורש.

שימו לב: מעבר הבדיקות שבאתר לא מהווה הבטחה לכך שתעברו את הבדיקות של הבודק האוטומטי! האתר מריץ את הקוד שלכם רק על מספר בדיקות מצומצם, בבדיקה האוטומטית הקוד יורץ על בדיקות אלו ומס' בדיקות נוספות. **לכן – כתבו בדיקות משלכם על-מנת לוודא כי הקוד שלכם נותן את הפלט המצופה בכמה שיותר מקרים!**

כאמור, באתר הקורס מסופקים לכם קבצי קלט ופלט מצופה עבור הבדיקות, על-מנת שתוכלו להשתמש בהם לביצוע diffMerge במקרה שהאתר אומר שאתם לא עוברים בדיקה מסויימת. **פתחו אותם וודאו שאתם מבינים מדוע הפלט הוא הנכון עבור אותו קלט.**

שאלות ותשובות נפוצות בנוגע לתרגילי הבית יתפרסמו באתר כל כמה זמן תחת סעיף -FAQ – **חובה להיכנס ולהתעדכן מדי פעם! כל דגש שמפורסם שם הינו מחייב!**

בהצלחה!