מבוא לתכנות מערכות 10010 סמסטר ב' – תשפ"א

תרגיל בית מס' 1

נושא התרגיל: מטריצות מצביעים

יש להגיש אך ורק דרך תפריט המטלות שבאתר הקורס.

הנחיות הגשה כלליות:

- התרגיל ייבדק בסביבת Ubuntu
- הקוד חייב לעבור קומפילציה, קוד שאינו מתקמפל לא ייבדק.
- **יש להגיש קוד ללא הערות קומפילציה, warnings, קוד בו יהיו הערות יגרור הורדה של 10 נקודות.**
- יש להגיש את תיקיית הקוד, מקובצת לקובץ אחד **ששמו כשם הסטודנט**. שם פרטי ומשפחה. אין להגיש פרויקט ב eclipse אלא קבצי c. h. ו Makefile
- ב Makefile חייבת להיות אפשרות ל clean
- ניתן לעבוד בזוגות - **במידה ומגישים בזוג קובץ ההגשה יהיה מורכב משמות 2 המגישים ושני המגישים צריכים לעלות את העבודה למודל.**

הוראות כלליות:

1. יש להקפיד על כללי הנדסת התוכנה:
   1. פונקציה לא אמורה להיות ארוכה מ 25 שורות. אם יש קטע קוד שניתן לתת לו כותרת מה הוא מבצע יש לכתוב אותו בפונקציה נפרדת.
   2. יש לחלק את הקוד לתת פונקציות מתאימות. אין לשכפל קוד. קוד זהה אמור להיות בפונקציה מתאימה.
   3. **יש לחלק כל שאלה לקבצי c ו h נפרדים**, יש לכתוב פונקציות כלליות בקובץ נפרד.
   4. מבנה התכנית (הזחות) ותיעוד במידת הצורך .
   5. חובה להשתמש בקבועים במקומות המתאימים .
   6. יש להשתמש בפונקציות קצרות, כלליות, קריאות ושימושיות.
   7. יש להקפיד על בדיקת תקינות קלט, אפשר להניח שאם ביקשו מהמשתמש מספר הוא הכניס מספר אך יתכן ולא בטווח הנכון.
   8. הפלט צריך להיות כפי שניתן בתרגיל במידה וניתן.
   9. קוד קצר, לא מסורבל ויעיל הן מבחינת כתיבתו והן מבחינת ריצת התוכנית.

בתרגיל בית זה יש 3 משימות.

יש לכתוב תכנית שמציגה תפריט המאפשר למשתמש לבחור את המשימה אותה הוא רוצה להריץ וכן תהיה אפשרות לצאת. התפריט הראשי יוצג כל עוד לא בחרו לצאת. כלומר בסיום כל משימה יוצג התפריט הראשי שוב. בחירת E/e תצא הודעה למסך "Bye Bye" ונצא מהתוכנית.

התפריט הראשי

Please choose one of the following options

S/s – Sudoku

B/b - Biggest Matrix Sum

N/n - Number Game

"E/e - Quit

להצגת התפריט יש להשתמש בתבנית switch, יש לדאוג שהמשתמש יוכל להקליד אות קטנה או גדולה.

עבור כל אפשרות בתפריט יש לכתוב פונקציה נפרדת, אין קוד ב switch למעט **קריאה** לפונקציה זו.

אין להשתמש בסוגרים מרובעים [ ] באף אחת מפונקציות בתרגיל למעט בתחילת הפונקציה הראשית של כל תרגיל שם מגדירים את המטריצה הראשונית. אין להגדיר מערכי עזר למעט מערך עזר אחד בפונקציה הראשית של משימת הסודוקו.

משימה 1 - Sudoku:

כתוב תוכנית שבודקת נכונות של לוח תשבץ סודוקו. בגודל N*N כך ש √ N מספר שלם.

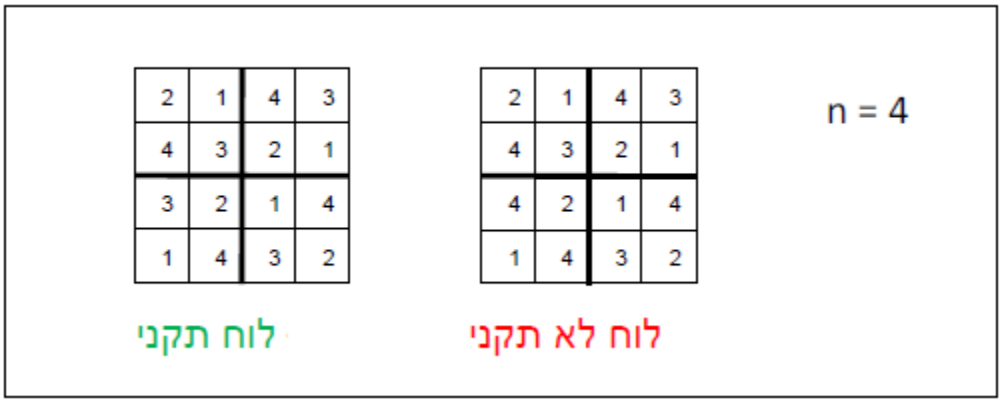
המטריצה תייצג את לוח הסודוקו.

את הבדיקות יש לבצע ע"י מעבר בודד על כל שורה, עמודה ותת ריבוע. אין להשתמש ב- [ ] באף פונקציה. רק בהגדרת לוח המשחק בפונקציה הראשית של המשימה. בפונקציה זו גם ניתן להגדיר עוד מערך עזר אחד לצורך עבודה יעילה.
את לוח המשחק הראשי נגדיר בגודל מקסימאלי של N*N אך בפועל המשתמש יבחר גודל לוח משחק החייב להיות שווה או קטן מ N, וכן מספר שיש לו שורש ריבועי.

מהלך checkSudoku :

- קלוט מהמשתמש גודל לוח תקין לסודוקו.
- אתחל את המטריצה תוך קליטת נתונים מהמשתמש.
- הצג את המטריצה.
- כתוב פונקציה אשר מקבלת את הלוח ומחזירה האם הלוח תקין.
  לוח סודוקו תקין אם כל השורות, עמודות וריבועים תקינים.

יש לחלק את העבודה לפונקציות עזר לפחות 5. (לי יש 8)



רמז: פונקציה memset המוגדרת string.h יכולה לשמש אתכם לאיפוס מערך העזר.

משימה 2 - Biggest Matrix Sum:

במשימה זו נחפש תת מטריצה בתוך מטריצה גדולה שסכום אבריה מקסימאלי.

- דוגמא: עבור מטריצה 20X20 אשר מחפשים בה תת מטריצה בגודל 2X2 נמצאה המטריצה המודגשת בצהוב. **שים לב: הדוגמא אומנם מתייחסת למטריצות מרובעות אך הדבר לא מחויב.**

50	50
50	49

23	35	40	-2	1	3	-28	-34	-45	46	-16	35	-21	29	-22	-24	17	-7	-4	13
-13	-20	11	21	-31	43	8	17	-40	0	47	-15	9	-45	47	13	11	22	-47	-36
34	4	-45	-12	-28	-10	36	32	-24	27	-49	50	50	-18	21	27	4	-15	-22	40
-11	-20	43	-39	13	-4	33	25	7	44	24	-24	-4	-13	-4	8	26	2	19	-16
18	-24	42	-30	15	-4	-23	19	-42	-10	-47	-44	-19	47	-45	-19	-32	15	8	20
-30	13	42	21	-35	-25	-21	13	-47	36	-26	6	16	48	24	43	40	-5	36	23
-48	48	-25	3	32	-36	3	-46	50	50	32	-25	33	-47	48	42	21	-37	-42	-46
-4	46	36	20	-36	48	50	4	50	49	14	19	-18	-21	10	-22	-2	-45	-6	15
-18	17	3	42	-24	-37	7	4	-46	-6	12	41	11	-43	17	-2	-26	17	38	38
50	7	-22	40	-20	-21	21	-19	3	46	35	1	-26	-20	23	19	20	28	-25	-3
-17	21	-9	20	28	21	-12	5	38	43	9	-27	25	-49	37	17	22	-21	-46	19
10	-19	-6	16	34	-32	-15	-46	24	-12	5	41	-10	35	21	-44	-13	18	16	-42
-37	14	12	-36	8	23	9	26	-15	0	-6	22	-32	-4	13	-43	21	9	-35	27
-24	-41	42	1	37	-1	34	48	30	0	9	-6	-44	31	10	-37	28	16	46	-9
20	16	-23	8	33	35	-41	-8	-42	11	19	18	45	36	32	-24	37	24	21	36
-46	38	-9	28	40	11	19	50	-20	50	-19	24	3	20	47	9	21	7	-17	-24
49	-17	2	40	2	-29	3	-22	16	40	3	41	43	-46	1	-17	4	6	44	-22
-25	20	5	50	1	-2	-26	40	10	4	-50	-29	18	49	-30	6	-43	-48	-18	-28
-25	15	-27	-22	46	19	17	45	49	43	15	-31	-31	-23	-24	-27	-3	-29	-26	19
-40	17	-20	-13	-29	-49	25	24	-3	-5	48	13	39	14	30	47	44	-45	30	-7

במשימה יש לכתוב לפחות פונקציה אחת שמחזירה בעזרת מצביעים יותר מנתון בודד.

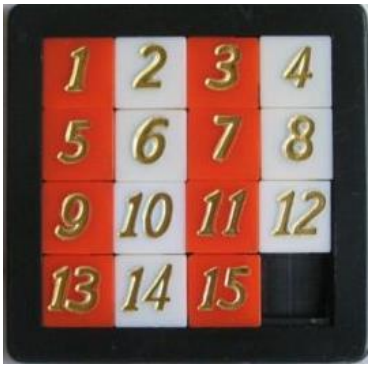
המשימה מבצעת את הפעולות הבאות: שים לב כל שלב הוא לפחות פונקציה אחת נפרדת

1. הגדר מטריצה בגודל ROWS שורות ו COLS עמודות בחר ערכים כרצונך. *הדפס ערכים אילו למסך.*
2. קלוט מהמשתמש שני מספרים אשר יגדירו את גודל המטריצה האפקטיבי כך שמספר השורות יהיה בין 1 ל ROWS כולל ומספר העמודות יהיה בין 1 ל COLS כולל. *הדפס ערכים אילו למסך.*
3. צור את המטריצה

משלב זה אין להשתמש ב סוגריים [ ] ואין להגדיר עוד מטריצה.

4. אתחל את המטריצה במספרים אקראיים, בחר טווח כרצונך, הגדר כמובן בקבועים. *הדפס ערכים אילו למסך.*
5. קלוט מהמשתמש שני מספרים נוספים אשר יגדירו גודל תת המטריצה *הדפס ערכים אילו למסך.* (שים לב בקלט שגודל השורות והעמודות בתת המטריצה צריך להיות קטן מהשורות והעמודות במטריצה הגדולה).
6. חפש בתוך המטריצה הגדולה תת מטריצה (לפי הגודל שנבחר) אשר סכום איבריה הוא המקסימלי . **יש לקחת בחשבון רק מקומות שהמטריצה הקטנה נכנסת במלואה במטריצה הגדולה.**
7. הדפס את מטריצה המספרים הגדולה ואת מטריצת המקסימום.
8. במידה ויש יותר מתת מטריצה אחת עם סכום מקסימאלי הדפס את הראשונה שמצאת.

משימה 3 - פאזל סידור מספרים:



פאזל המספרים הוא משחק על לוח בגודל כלשהו N*M עליו רשומים N*M מספרים מ-0 עד N*M-1 לאחר שערבבו היטב. המשבצת הנותרת נשארת ריקה. **0 מציין משבצת ריקה.**

דוגמא: עבור גודל 4*4 בתחילת המשחק המספרים יכולים להיות מסודרים באופן הבא.

13	4	8	9
11	7	1	5
6	2	10	12
3	15	14	0

מטרת המשחק היא לסדר את הלוח בסדר עולה של המספרים רק באמצעות הזזת של מספרים ממשבצות סמוכות למשבצת הריקה לתוך המשבצת הריקה. למשל בלוח שלמעלה אפשר לבצע שתי הזזות : 14 ימינה, 12 למטה. המשתמש בוחר את המהלך. במידה והמשתמש בחר להעביר את 12 למטה הסדר החדש יראה כך:

13	4	8	9
11	7	1	5
6	2	10	0
3	15	14	12

המשחק מסתיים כאשר הלוח מסודר :

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	0

במשימה יש לכתוב לפחות פונקציה אחת שמחזירה בעזרת מצביעים יותר מנתון בודד.

לדוגמא: פונקציה המחפשת מספר מסוים במטריצה ומחזירה את הקואורדינטות של השורה והעמודה בה המספר נמצא. הפונקציה מחזירה 0 אם המספר לא נמצא 1 אם נמצא.

משימת התרגיל:

1. יש לאתחל מטריצה בגודל N*M ב מספרים מ 0 עד N*M-1 , 0 יציין מקום ריק.
2. יש להעביר את המטריצה תהליך של ערבוב מספרים. בצע זאת באופן הבא:
   - הגרל כיוון: מעלה, מטה, ימינה, שמאלה.
   - במידה והמהלך חוקי החלף את ה 0 בתא שנבחר.
   - בצע מספר קבוע, SHUFFEL_COUNT , החלפות שכאילו כדי לערבב את המטריצה.

   נניח 10.

נניח שבמהלך הערבוב הראשון הגרלתי כיוון מעלה אזי המטריצה תראה:

1	2	3	4
5	6	7	8
9	15	11	0
13	14	10	12

3. עכשיו מתחיל המשחק, המשתמש ינסה לסדר את לוח המשחק.

   בכל שלב מבצעים את הפעולות הבאות:

   1. בדיקה האם הלוח מסודר בסדר עולה – אם כן הכרז על ניצחון של המשתמש וסיים את התוכנית.

      התוכנית תכתוב: You win! The game is over!
   2. אם לא שאל את המשתמש איזה מספר הוא רוצה להזיז למשבצת הריקה (Your step:).
   3. אם המשתמש הקיש מספר לא חוקי (מספר שלא סמוך למשבצת הריקה או מספר מחוץ לתחום יש להדפיס הודעת שגיאה ולחזור לשלב 2 (Invalid step!).
   4. ביצוע הזזה. (שים לב שאחרי ההזזה ידוע איפה ה 0 נמצא – אין לחפש שוב!)
   5. הדפסת לוח על המסך.

דוגמה: תגובות המשתמש מסומנות באדום, המשחק עוד לא הסתיים

```
13    4    8    9
11    7    1    5
 6    2   10   12
 3   15   14    0

Your step: 10
Invalid step!
Your step: 12

13    4    8    9
11    7    1    5
 6    2   10    0
 3   15   14   12
```

בהצלחה