

## **בדיקה ידנית:**

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

- 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.
- רוחב שורה - רוחב כל שורה (כולל הערות והזחות) לא יעלה על 75 תווים. ניתן לסמן אורך של שורה בקודבלוקס (ראו מסמך מצורף).

- 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 מטרותם.
- 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 (גם עבור הפונקציות שמוגדרות בתרגיל 1).
  - 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.

## שאלה 1: מבוא לפונקציות

א. איבר במטריצה נקרא חזק אם הוא גדול מכל שכניו (מלמעלה, מלמטה, מימין ומשמאל).  
כתבו פונקציה שחתימתה:

```
int is_strong (int mat[N][N], int row, int column)
    המקבלת מטריצה mat, אינדקס שורה row ואינדקס עמודה column, ומחזירה 1 אם
    mat[row][column] הוא חזק, ואחרת מחזירה 0.
```

ב. במטריצה, דרגת הריווח של איבר x מוגדרת להיות ההפרש של x משכנו הימני y, בערך מוחלט (|x-y|). אם x אין שכן ימני, דרגת הריווח שלו תהיה 0.

כתבו פונקציה שחתימתה:

```
int space_rank (int mat[N][N], int row, int column)
    המקבלת מטריצה mat, אינדקס שורה row ואינדקס עמודה column, ומחזירה את דרגת הריווח של
    mat[row][column].
```

ג. דרגת הריווח של מטריצה מוגדרת להיות סכום דרגות הריווח של כל איבריה.  
כתבו תוכנית (main) הקולטת מהמשתמש איברים (מספרים שלמים) של מטריצה 4x4 **ומחזרת**  
ומדפיסה את מספר האיברים החזקים במטריצה, ואת דרגת הריווח של המטריצה.

הערות למימוש:

- במימוש הפונקציות בסעיפים א'-ב' אין להשתמש במשפטי תנאי או בלולאות. אם מימשתם פונקציות עזר שנקראות מתוך פונקציות אלה, עליהן לעמוד בדרישה זו גם כן.
- אין צורך לבדוק את תקינות הקלט בשאלה זו, כלומר מובטח שהקלט מכיל 16 מספרים שלמים בטווח הייצוג של int.
- במימוש הפונקציות בסעיפים א'-ב' אפשר להניח שמתקיים  $row < N$ ,  $column < N$ .
- ההגדרה של שכן אינה ציקלית: איבר מהשורה הראשונה אינו שכן של איבר מהשורה האחרונה, ואיבר מהעמודה הראשונה אינו שכן של איבר מהעמודה האחרונה.
- יש לקלוט את ערכי המטריצה שורה אחרי שורה, משמאל לימין.
- יש להיעזר בפונקציות מסעיפים א'-ב' בכתיבת התוכנית בסעיף ג'. חישוב חוזק או דרגת ריווח של איבר במטריצה יבוצע אך ורק ע"י פונקציות אלה (ופונקציות הנקראות מתוכן, אם מימשתם כאלה).
- אין להשתמש בספרייה math.h בשאלה זו (חשבו כיצד לממש ערך מוחלט ללא שימוש בתנאים).
- בשאלה זו יש להניח  $N=4$  (ולהגדיר זאת ב-define).

דוגמאות הרצה:

```
Please enter a matrix:
0 0 1 1
0 0 1 1
0 0 1 1
0 0 1 1
Strong elements: 0
Space rank: 4
```

```
Please enter a matrix:
0 3 0 3
3 0 3 0
0 3 0 3
3 0 3 0
Strong elements: 8
Space rank: 36
```

בדוגמה השמאלית: אין איברים חזקים במטריצה. דרגת הריווח של כל האפסים בעמודה השנייה היא 1, ולכן דרגת הריווח של המטריצה היא 4.

בדוגמה הימנית: כל האיברים בעלי הערך 3 הם חזקים. דרגת הריווח של 12 האיברים שאינם בעמודה האחרונה היא 3, ולכן דרגת הריווח של המטריצה היא 36.

## שאלה 2: מערכים חד מימדיים ופונקציות

ברוצנו לכתוב תוכנית המחשבת סטטיסטיקות עבור ציונים במבחן ולממש שיטה לעדכון הציונים תוך שמירה על ממוצע וסטיית תקן (Standard Deviation) מקסימלית נתונים.

ממשו את הפונקציות הבאות:

1. פונקציה בשם **minMaxGrades** המקבלת כפרמטר מערך של ציונים ממשיים, מספר הציונים בו, ומשתנה בוליאני. מחזירה את הציון המינימלי במערך אם ערך המשתנה הבוליאני שווה ל- `true`, אחרת מחזירה את הציון המקסימלי במערך.
2. פונקציה בשם **meanGrades** המקבלת כפרמטר מערך של ציונים ממשיים ומספר הציונים בו, ומחזירה את ממוצע הציונים במערך.
3. פונקציה בשם **stdGrades** המקבלת כפרמטר מערך של ציונים ממשיים, מספר הציונים בו וממוצע הציונים, ומחזירה את סטיית התקן של הציונים במערך.

סטיית התקן של  $N$  מספרים  $x_1, \dots, x_N$  מחושב באופן הבא:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

כאשר  $\sum_{i=1}^N (x_i - \bar{x})^2$  מייצג סכום מ-1 עד  $N$  של המספרים  $x_i - \bar{x}$  ו- $\bar{x}$  מייצג הממוצע של  $x_1, \dots, x_N$ .

4. פונקציה בשם **printGrades** המקבלת כפרמטר מערך של ציונים ממשיים ומספר הציונים בו, ומדפיסה את כל הציונים במערך עד ספירה אחת לאחר הנקודה העشرונית, כולם מופרדים על ידי **רווח** ובסוף מדפיסה **ירידת שורה**.
5. פונקציה בשם **updateGrades** המקבלת כפרמטר מערך של ציונים ממשיים, מספר הציונים בו, ממוצע המצופה (ייתכן שיהיה שונה ממוצע הציונים לפני העדכון) וסטיית תקן מקסימלית. הפונקציה מעדכנת את הציונים במערך בשני שלבים:
  - אם סטיית התקן של הציונים (נסמן אותה ב- $\sigma_{cur}$ ) גדולה מסטיית התקן המקסימלית (נסמן אותה ב- $\sigma_{max}$ ), אז נשנה כל ציון מ- $x_i$  ל- $x_i \cdot \frac{\sigma_{max}}{\sigma_{cur}}$ .
  - אם ההפרש בערך המוחלט בין ממוצע הציונים לאחר שלב 1 (נסמן אותו ב- $\mu_1$ ) מהממוצע הנתון כפרמטר לפונקציה (נסמן אותו ב- $\mu_2$ ) גדול מ-1, אז נשנה כל ציון מ- $x_i$  ל- $x_i - \mu_1 + \mu_2$ .

בנוסף, כתבו תוכנית אשר:

1. מדפיסה את הודעת הפתיחה בעזרת הפונקציה `void printGradesOpening()`.
2. מקבלת מהשתמש הציונים של הנבחנים, ואחר כך ממוצע הציונים הדרוש לאחר עדכון הציונים (עד סיום קלט).
3. מדפיסה את הציונים הישנים בעזרת הפונקציה `void printOldGrades(...)` ואחר כך הסטטיסטיקות בעזרת `void printOldGradesStatistics(...)`.
4. מעדכנת את הציונים לפי הממוצע שהשתמש סיפק וסטיית תקן **מקסימלית** קבועה ששווה ל-10.
5. מדפיסה את הציונים החדשים בעזרת הפונקציה `void printNewGrades(...)` ואחר כך הסטטיסטיקות בעזרת `void printNewGradesStatistics(...)`.

### דגשים:

1. **ניתן להניח** שמספר הנבחנים הוא לכל היותר 50, וניתן להניח שהשתמש יכניס לפחות ציון אחד ולא יכניס יותר מ-50 ציונים.
2. **ניתן להניח** שהקלט תקין, כלומר המשתמש יכניס מספרים ממשיים אי שליליים (בין 0 ל-100).
3. **ניתן להניח** שציונים המעוכדנים יהיו בין 0 ל-100.
4. פונקציות ההדפסה (חוץ מ-**printGrades**) מסופקות בקובץ `hw2q2_prints.c`, תעתיקו אותן לקובץ הפתרון. שימו לב שחלק מפונקציות המסופקות בקובץ משתמשות בפונקציה **printGrades**.
5. מותר להשתמש אך ורק בפונקציה `sqrt` בספרייה `<math.h>`.
6. **מותר ומומלץ** לממש פונקציות עזר אחרות.

### שאלה 3: הדפסת היסטוגרמה

בשאלה זו עליכם לכתוב תוכנית הקולטת מהמשתמש את ציוני הסטודנטים בקורס "מבוא למדעי המחשב", מיקום ציר ההיסטוגרמה (U/D/R/L) ומדפיסה היסטוגרמה שלהם במקביל לציר שלה (היסטוגרמה מכילה עבור כל ציון אפשרי את מספר הסטודנטים שקיבלו אותו). בגרסה זו של הקורס, הציונים הינם מספרים שלמים בין 1 ל-9 (כולל). כמו-כן, התוכנית קולטת מהמשתמש את התו באמצעותו תוצג ההיסטוגרמה.

הערות למימוש:

- יש לקלוט ציונים עד לקבלת EOF מהמשתמש (בפרט, מספר הסטודנטים אינו ידוע מראש, אך מובטח שהוא לא חורג מטווח הייצוג של int).
- יש לבדוק אם המשתמש לא מכניס מיקום ציר חוקי ולהדפיס הודעה כמתואר בדוגמאות ולבקש חזרה להכניס מיקום הציר.
- אין צורך לבדוק את תקינות הקלט בשאר במקרים: ראשית מתקבל תו כלשהו (ללא מגבלות), אח"כ מתקבל מיקום ציר ההיסטוגרמה שצריך לבדוק, אח"כ מתקבלים מספרים בטווח הרצוי, ולבסוף מתקבל EOF.
- המספרים שעל הציר ההיסטוגרמה מייצגים את הציונים האפשריים. כל סטודנט מיוצג ע"י התו הנבחר, הממוקם בעמודה/שורה המתאימה לציונו (בדוגמה השמאלית למשל, ישנן 4 כוכביות בעמוד המייצג ציון 4, המייצגות 3 סטודנטים שקיבלו ציון זה).

דוגמאות הרצה (ctrl+Z מדמה EOF):

Please enter a character:

\*

Please enter histogram axis:(U/D/R/L)

T

Wrong direction, try again!

Please enter histogram axis:(U/D/R/L)

D

Please enter grades:

1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 6 6 6 6 7 7 7  
8 8 9

^Z

Histogram:

\*

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \*

1 2 3 4 5 6 7 8 9

Please enter a character:

O

Please enter histogram axis (U/D/R/L)

L

Please enter grades:

6 7 8 9 1 2 3 4 5 5

^Z

Histogram:

1 O

2 O

3 O

4 O

5 O O

6 O

7 O

8 O

9 O

דגשים להדפסת ההיסטוגרמה:

1. יש להכניס רווח בין כל שתי עמודות (אך אין להכניס רווח לפני העמודה הראשונה או אחרי האחרונה).
2. נסמן את הציון הנפוץ ביותר ב- $m$ , ואת מס' הסטודנטים שקיבלו אותו ב- $C(m)$ . עבור ציון כלשהו  $i$  אותו קיבלו  $C(i)$  סטודנטים, ב- $C(i)$  השורות העמודות הכי קרובות לציר ההיסטוגרמה של ליד הציון  $i$  יש להדפיס את התו המיוחד, ובשאר  $C(m)-C(i)$  העמודות הרחוקות מהציר יש להדפיס רווחים.
- למשל, בדוגמה השמאלית לעיל, היות שיש רק סטודנט אחד קיבל ציון 1, בעמודה זו הודפס תו אחד \* וארבעה רווחים מעל. היות ושלושה סטודנטים קיבלו את הציון 7, בעמודה זו הודפסו 3 כוכביות ושני רווחים.
3. יש להדפיס בהתאם לפורמט המוצג בדוגמאות.