**מרתון מל"מ**

**רשימות :**

שתי דרכים עיקריות לרוץ על רשימות :

בהנחה ש list היא רשימה :

For i in range (start , end , step) :

List[i]

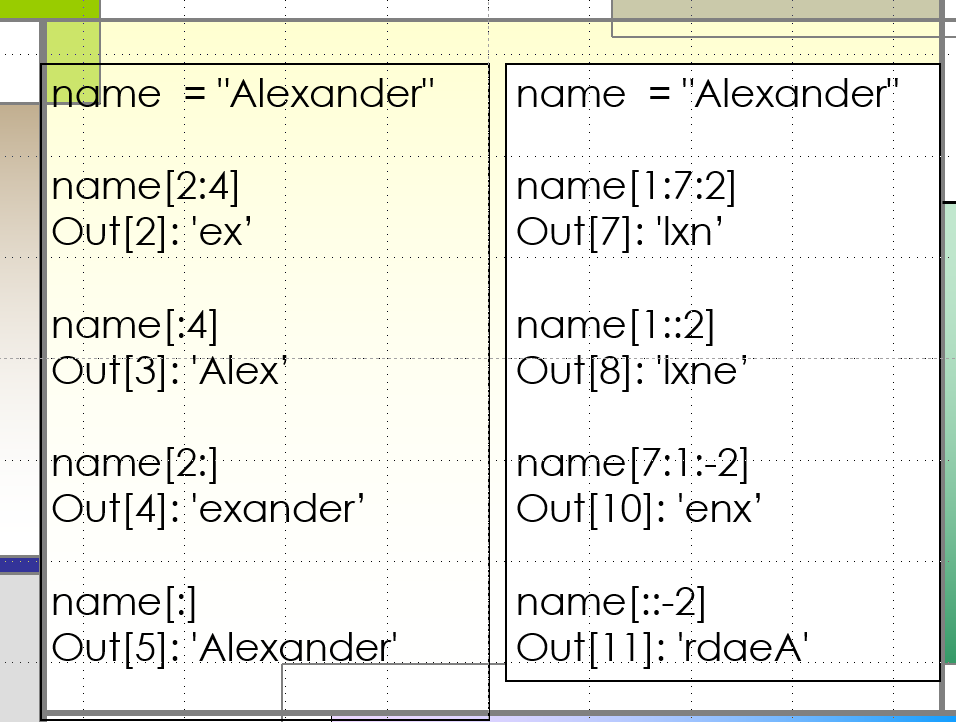
For item in list :

Item

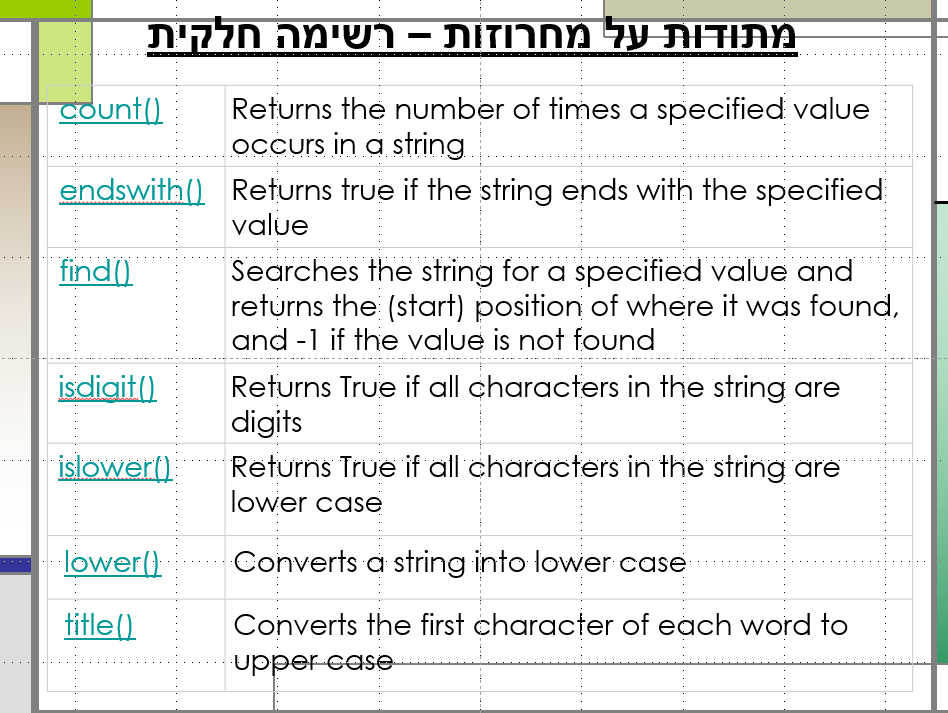
* פונקצית len מחזירה את אורך הרשימה .
* פונקציית range לא מגיעה אף פעם לערך השני שלה כלומר עוצרת איטרציה אחת לפני ה end .
* **כשמבקשים סיבוכיות O(n) שווה** **לחשוב על רשימת מונים.**

**מחרוזות :**

**פעולות נפוצות על מחרוזות :**

****

* **מחרוזות אינם ניתנות לשינוי , כלומר יש רק קריאה ממחרוזת ולא כתיבה !**

****

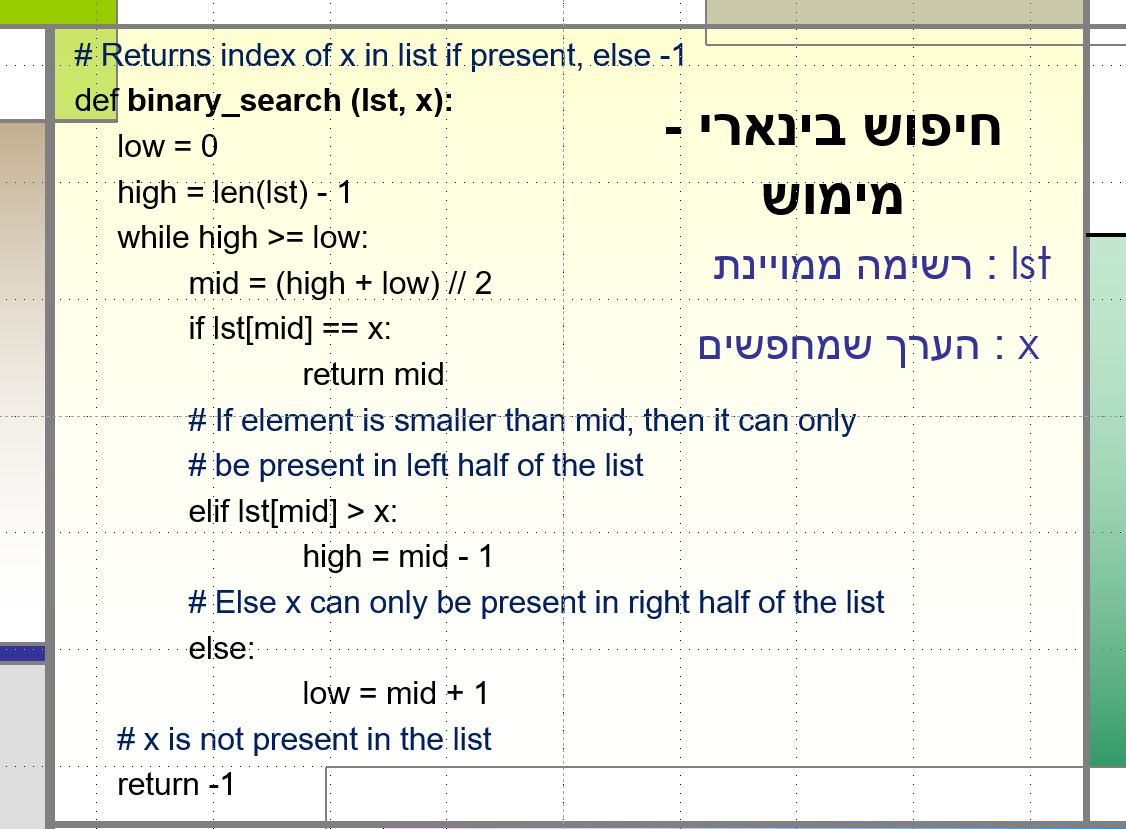
* **ההבדל בין אות גדולה לאות קטנה (אותו סוג ) הוא 32 בטבלת האסקי .**

**סיבוכיות זמן ומקום**

**סיבוכיות זמן :** מספר הפעולות כפונקציה של אורך הסדרה

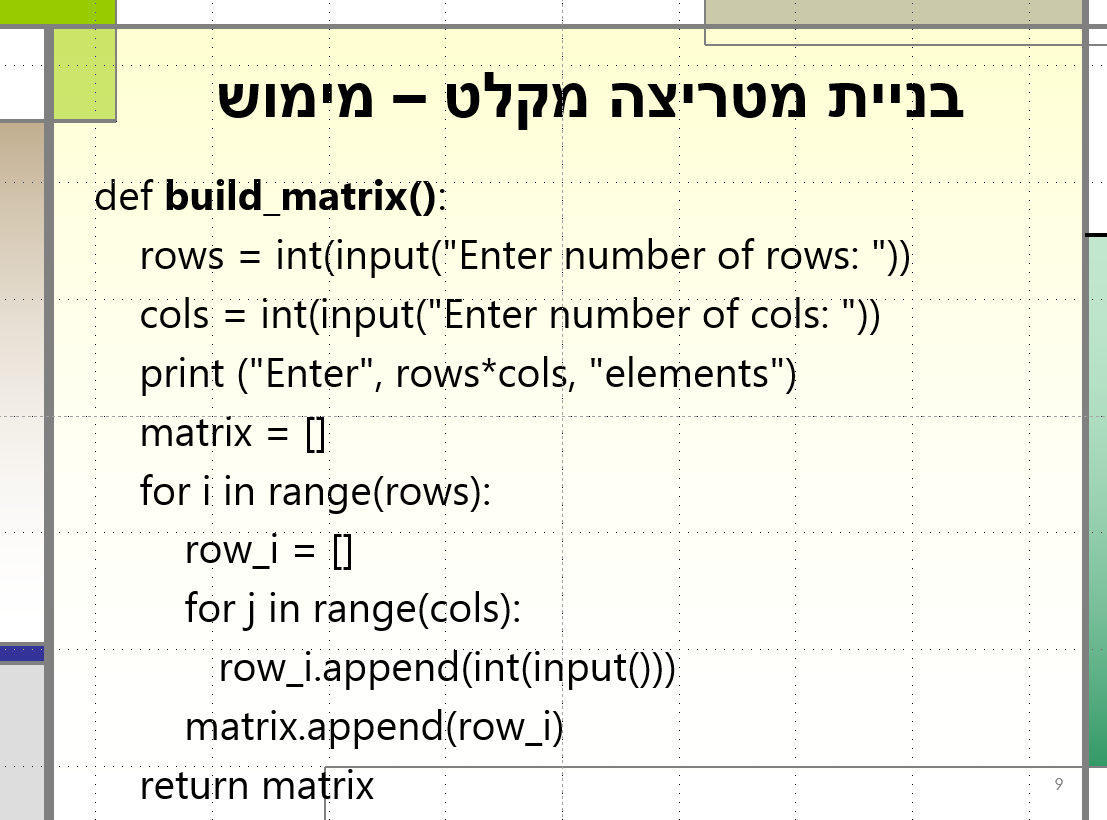
**סיבוכיות מקום :** מסתכלים רק על הזיכרון הנוסף הנדרש לביצוע האלגוריתם (למשל, רשימת/מחרוזת עזר), ולא כוללים את הפרמטרים המועברים לפונקציה.

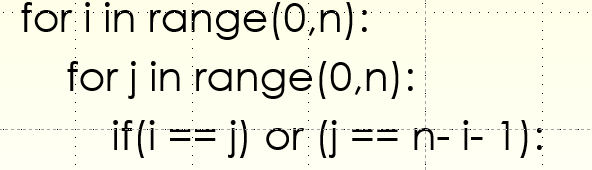
**חיפוש בינארי**

****

* **סיבוכיות הזמן של חיפוש בינארי הוא O(log n )**

**מטריצה**

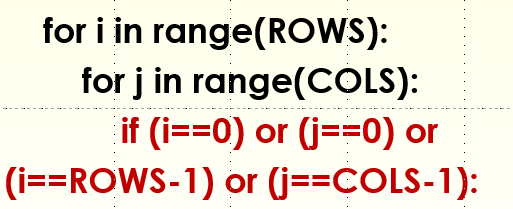
****

**מעבר על אלכסוני המטריצה :**

אלכסון משני

אלכסון ראשי

**מעבר על מסגרת מטריצה :**

****

**מבחנים**

**חורף 2020 מועד א'**

**שאלה 2 (24 נקודות)**

נגדיר **הזזה מעגלית** של רשימה, כהזזה של כל איבר ברשימה אל התא הנמצא מימינו, כאשר האיבר האחרון ברשימה מוזז לתא הראשון.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20 | 16 | 11 | 10 | 9 | 6 | 5 | 0 | -3 | -5 |

למשל, עבור הרשימה:

תוצאת ה**הזזה מעגלית** שלו תהייה:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16 | 11 | 10 | 9 | 6 | 5 | 0 | -3 | -5 | 20 |

נגדיר **הזזה מעגלית-K** של רשימה, כביצוע **k** פעמים **הזזה מעגלית** על הרשימה.

למשל, עבור הרשימה העליונה,

תוצאת **הזזה מעגלית-K** עבור k=3 תהייה הרשימה הבא:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 10 | 9 | 6 | 5 | 0 | -3 | -5 | 20 | 16 | 11 |

1. (8 נקודות) כתבו פונקציה **circle\_shift** המקבלת רשימת שלמים ar ומבצעת **הזזה מעגלית** של הרשימה.
2. (8 נקודות) כתבו פונקציה **circle\_shift\_K** המקבלת רשימת שלמים ar ומספר טבעי k<n, ומבצעת **הזזה מעגלית-K** של הרשימה.
3. (8 נקודות) כתבו פונקציה בוליאנית **is\_circle\_shift** המקבלת שני רשימות מטיפוס שלם באורך שווה ar1 ו- ar2 , ואת אורכם n, ומחזירה 1 אם ar2 הוא תוצאה של **הזזה מעגלית-K** (עבור k טבעי כלשהו) של הרשימה ar1, ו- 0, אחרת. יש להשתמש בפונקציה אחת לפחות מבין אלו שהוגדרו בשני הסעיפים הקודמים.

2.

1. (13 נקודות) כתבו פונקציה רקורסיבית **bigChar** המקבלת מחרוזת (טקסט), ומחזירה את אינדקס האות הראשונה שהיא אות גדולה. במקרה שלא קיימת אות גדולה במחרוזת, יוחזר -1.

עבור המחרוזת “When I am fine, I look fine” יוחזר: 0, כי האות הגדולה מופיעה בתחילת המחרוזת.

עבור המחרוזת “don't feel Like It!” יוחזר 11, כי האות הגדולה הראשונה היא L ונמצאת באינדקס ה- 11.

עבור המחרוזת “what can i do ?” יוחזר -1 , כי לא קיימת אות גדולה.

1. (4 נקודות) מהי סיבוכיות הזמן והמקום של הפונקציה שכתבתם בסעיף ב'? נמקו את קביעתכם. (לא יינתן ניקוד על תשובה לא מנומקת).

3.

נגדיר **ריבוע שלם**, כמטריצה ריבועית מסדר nXn המכילה את כל המספרים השלמים מ- 1 עד n2, n מספר טבעי. לדוגמה: כל אחת משתי המטריצות הבאות היא **ריבוע שלם** (מסדר 2X2 ומסדר (4X4:

|  |  |
| --- | --- |
| 1 | 4 |
| 2 | 3 |

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

1. (14 נקודות) כתבו פונקציה בשם **completeSq** המקבלת שני פרמטרים: מטריצת שלמים mat מסדר nXn, ואת n, ומחזירה 1 אם המטריצה היא **ריבוע שלם**, ו- 0 אחרת.

**דרישות סיבוכיות הזמן: O(n2).**

1. (2 נקודות) מהי סיבוכיות המקום של הפונקציה **completeSq** שכתבתם? נמקו את קביעתכם. (לא יינתן ניקוד על תשובה לא מנומקת).
2. (11 נקודות) כתבו פונקציה **findNum** המקבלת **ריבוע שלם** sq מסדר nXn ומחזירה את המיקום (מספר שורה ומספר עמודה) של המספר n2 .

(ניתן להניח שהפונקציה **findNum** אכן מקבלת **ריבוע שלם**.)

**מבחן 2019 מועד אביב**

**שאלה 1 (26 נקודות)**

1. (14 נקודות) כתבו פונקציה איטרטיבית (לא רקורסיבית) המקבלת שני רשימות של מספרים שלמים: רשימהa[] ו- רשימה b[], שני הרשימות ממויינים בסדר **לא יורד** (כלומר, כל איבר ברשימה גדול מקודמו או שווה לו). הפונקציה מדפיסה בשורה את כל המספרים שבשני המערכים בסדר **עולה ממש** (כלומר, כך שכל מספר גדול מקודמו. כל מספר יודפס פעם אחת בלבד).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 15 | 9 | 6 | 6 | 3 | 1 |

למשל, עבור שני המערכים הבאים:

|  |  |  |  |
| --- | --- | --- | --- |
| 6 | 5 | 3 | 3 |

תודפס שורת המספרים: 15 1 3 5 6 9

סיבוכיות הזמן הנדרשת היא O(n+m).

1. (10 נקודות) כתבו פונקציה רקורסיבית המקבלת רשימה של מספרים שלמים , a[]הרשימה ממוין בסדר לא יורד, ומדפיסה בשורה את כל המספרים ברשימה בסדר **עולה ממש** (כלומר, כך שכל מספר גדול מקודמו. כל מספר יודפס פעם אחת בלבד).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 15 | 9 | 6 | 6 | 3 | 1 | 1 |

למשל, עבור הרשימה:

תודפס שורת המספרים:1 3 6 9 15

1. (2 נקודות) רשמו את סיבוכיות הזמן והמקום של הפונקציה שכתבתם בסעיף ב'. נמקו תשובתכם.

**שאלה 2 (24 נקודות)**

1. (12 נקודות) נגדיר "תחילית" של מחרוזת כרצף של תווים (אחד אן יותר) המופיעים בתחילת המחרוזת.

כתבו פונקציה רקורסיבית *rec\_prefix* המקבלת שתי מחרוזות. מחרוזת s ומחרוזת pref . הפונקציה תחזיר את **אורך** התחילית הארוכה ביותר ***המשותפת*** לשתי המחרוזות.

למשל עבור ***abcdef***dvdf”"s[]= ו- pref[]="***abc***y" הפונקציה תחזיר 3 (כי התחילית הארוכה יותר המשותפת לשתי המחרוזות היא abc.

עבור **a**""s[]= ו- pref[]=”**a**by” הפונקציה תחזיר 1.

ועבור" "s[]= ו- pref[]="" הפונקציה תחזיר 0.

1. (12 נקודות) כתבו פונקציה איטרטיבית או רקורסיבית (לבחירתכם) cntMaxSub המקבלת שתי מחרוזות. מחרוזת s ומחרוזת sub . הפונקציה מחזירה את אורך התחילית הארוכה ביותר של sub המופיעה כרצף, **במקום כלשהו**, במחרוזת s .

למשל, עבור s[] = "xabsbe***abcdef***dvdf" ו- "***abcdef***g" sub[]= הפונקציה תחזיר 6.

,עבור s[] = "u***ab***acad" ו- "***ab***cdefg" sub[]= הפונקציה תחזיר 2

ועבור s[] = "" ו- "***ab***cdefg" sub[]= הפונקציה תחזיר 0.

**שאלה 3 (26 נקודות)**

1. (22 נקודות) כתבו פונקציה המקבלת מטריצה mat של **מספרים חד ספרתיים** אי-שליליים, ואת מספר השורות rows ומספר העמודות cols של המטריצה.

כותרת הפונקציה: void inAllRows (int \*\*mat, int rows, int cols)

הפונקציה תבדוק האם קיימים מספרים אשר חוזרים בכל שורה במטריצה, ואם יש כאלו היא תדפיס אותם, ללא חזרות.

למשל, עבור המטריצה הבאה מסדר 7X6 יודפס :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 7 | 1 | 3 | 0 | 4 | 6 |
| 2 | 5 | 0 | 6 | 6 | 1 |
| 6 | 1 | 7 | 2 | 4 | 0 |
| 1 | 1 | 7 | 6 | 8 | 9 |
| 5 | 5 | 6 | 1 | 5 | 3 |
| 4 | 1 | 4 | 3 | 6 | 0 |
| 4 | 6 | 1 | 7 | 4 | 3 |

Common element(s): 1 6

אם אין מספרים החוזרים בכל שורה במטריצה, יודפס: No common elements.

הערה: שימו לב שמספר עשוי לחזור כמה פעמים בשורה.

1. (4 נקודות) מהי סיבוכיות הזמן וסיבוכיות המקום של הפונקציה שכתבתם? נמקו את קביעתכם.

**מבחן שנת 2019 אביב מועד א'**

**שאלה 1 (28 נקודות)**

רשימה נקרת **ממוינת בסדר מעגלי עולה**, אם קיים ברשימה אינדקס i, i>0 , כך שכל איברי הרשימה החל מאינדקס זה ועד איבר אחד לפניו ממוינים בסדר עולה, כאשר האיבר שבא אחרי האחרון ברשימה הוא הראשון ברשימה.

לדוגמא, הרשימה הבאה **ממוינת בסדר מעגלי עולה**, עבור i=4, כי סדרת הערכים המתחילה ב- -5 ומסתיימת ב- 20 היא ממוינת בסדר עולה.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 9 | 6 | 5 | 0 | -3 | **-5** | 20 | 16 | 11 | 10 |

הערה לשני סעיפי השאלה: ידוע ואין צורך לבדוק שהרשימה שמקבלת הפונקציה ממוין בסדר מעגלי עולה ושערכיו שונים זה מזה.

* 1. (15 נקודות) כתבו פונקציה, המקבלת רשימה של מספרים שלמים, שונים זה מזה, **ממוין בסדר מעגלי עולה** . הפונקציה מחזירה את ערך האינדקס ברשימה אשר החל ממנו הרשימה ממוינת בסדר מעגלי עולה.

למשל, עבור הרשימה שבדוגמא יוחזר האינדקס **4**.

דרישות הסיבוכיות הן:

זמן O(log n), כאשר n הוא אורך הרשימה, ומקום O(1).

* 1. (13 נקודות) כתבו פונקציה המקבלת רשימת מספרים שלמים, שונים זה מזה, **ממוין בסדר מעגלי עולה** ואת אורכו. הפונקציה משנה את רשימה זו לממוינת בסדר עולה (לא מעגלי).

פונקציה זו צריכה להשתמש בפונקציה שהוגדרה בסעיף א'.

סיבוכיות הזמן של הפונקציה לא תהייה גדולה מ- O(n), כאשר n הוא אורך הרשימה.

לדוגמא, עבור הרשימה שלמעלה, יתקבל הרשימה הממוינת הבאה:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 20 | 16 | 11 | 10 | 9 | 6 | 5 | 0 | -3 | -5 |

**שאלה 2 (22 נקודות)**

כתבו פונקציה המקבלת מטריצה mat של **תווים**, את מספר השורות rows ומספר העמודות cols של המטריצה, ומחזירה את המספר השלם הגדול ביותר שניתן ליצור מן הספרות הנמצאות במטריצה. יש להניח כי במטריצה ישנו לפחות תו אחד שהוא ספרה.

כותרת הפונקציה: LargestNum (mat ,rows ,cols)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **7** | X | \* | # | a |
| **1** | T | P | X | # |
| # | m | **5** | @ | **2** |
| @ | \* | k | **2** | b |

למשל, עבור המטריצה הבאה מסדר 4X5 יוחזר הערך 75221.

(הערה: ניתן להניח שהמספר המחושב ומוחזר הוא בטווח הערכים של long int ).

**שאלה 3 (26 נקודות)**

"רצף זוגי" הוא רצף תאים ברשימה שבהם שמורים מספרים זוגיים. אורך הרצף יכול להיות 1 או יותר.

1. (13 נקודות) כתבו פונקציה **רקורסיבית** המקבלת רשימה שלמיםa .

הפונקציה מחזירה את מספר הרצפים הזוגיים ברשימה.

למשל, עבור הרשימה a[] = { 1,-3, **82**, 5, **6, 44** } הפונקציה תחזיר 2,

ועבור הרשימה a[] = { 1,-3, 17 } הפונקציה תחזיר 0.

1. (13 נקודות) כתבו פונקציה **איטרטיבית** (לא רקורסיבית) המבצעת בדיוק אותה משימה כמו הפונקציה שכתבתם בסעיף א. כלומר, מקבלת רשימת שלמיםa ואת אורכו n ומחזירה את מספר הרצפים הזוגיים ברשימה.