

תרגיל בית 2 מבוא למדעי המחשב מ/ח – 234114/7 – סימסטר אביב 2020.

שאלות על התרגיל יש לשלוח לדן ברקוביץ', לכתובת: danb2341147@gmail.com ולא לכתובת הטכניונית (המופיעה באתר הקורס).

1. **החומר הנדרש לפתרון התרגילים:** משפטי תנאי, לולאות, פונקציות ומערכים - כולל העברת מערכים כקלטים לפונקציות.
 - ככל הנראה העברת מערכים כקלטים לפונקציות טרם נלמד בהרצאות ובתרגולים בעת פרסום התרגיל (וילמד בשבועות הקרובים). באתר הקורס hw2 → assignment מופיע מסמך בשם passing arrays to function בו מופיע הסבר וכן דוגמה להעברת מערכים לפונקציה.

2. **ניתן להשתמש בספריית:** stdio, stdbool.

3. **אסור להשתמש ב:**

- משתנים סטטים וגלובלים (global, static variables).

- מערכים מאורך משתנה variable sized arrays.

- הקצאת זיכרון דינמית.

- כל ספריה שאינה stdio, stdbool.

4. **שימוש ב- define:** מלבד הקבועים 0 ו-1, יש להחליף כל קבוע באמצעות מנגנון ה- define. דוגמה, בקוד הבא משתמשים בקבועים: 9, 'z', 'a'.

```
#include <stdio.h>
int main(){
    char c;
    int x;
    scanf(" %c %d", &c, &x);
    if(c < 'a' || c > 'z'){
        printf("Error 1.");
    }
    if(x < 1 || x > 9){
        printf("Error 2.");
    }
    return 0;
}
```

נשתמש ב- define במקום הקבועים 9, 'z', 'a'.

```
#define FIRST_LET 'a'
#define LAST_LET 'z'
#define LAST_DIGIT 9
#include <stdio.h>
int main(){
    char c;
    int x;
    scanf(" %c %d", &c, &x);
    if(c < FIRST_LET || c > LAST_LET){
        printf("Error 1.");
    }
    if(x < 1 || x > LAST_DIGIT){
        printf("Error 2.");
    }
    return 0;
}
```

5. **ספירת שורות בפונקציות:** מספר השורות בכל פונקציה בתוכנית (כולל main) לא יעלה על 16 שורות (כלומר **מש-פחות** מ-17 שורות). באתר הקורס, תחת hw2 → assignment מופיע מסמך בשם line counting procedure המפרט כיצד לספור שורות, אנא קראו אותו. (לדוגמה, שורה המכילה רק סוגר מסולסל אינה נחשבת בספירת השורות).

6. **אורך שורה:** כל שורה בקוד לא תעלה על 150 תווים.
7. **הדפסות:** הסברים נוספים יופיעו בהרחבה בכל אחד מהתרגילים, אך כדי למנוע שגיאות הנובעות מהדפסת מחרוזות שאינן בתצורה המצופה, בקבצים hw2q1.c, hw2q2.c, hw3q3.c המופיעים באתר הקורס hw2 → assignment ממומשות פונקציות המדפיסות את הנדרש למסך לפי ההוראות שמופיעות בכל אחד מהתרגילים. מומלץ (אך אין זו חובה) להשתמש בפונקציות אלו לשם ההדפסות (העתיקו את המימושים מהקבצים המצורפים).
8. הגשה **בבודדים**. עליכם לכתוב את הפתרונות לבד ולהגיש ביחידים.
9. קראו את השאלות בעיון לפני שתתחילו בפתרון.
10. הקפידו לתעד את הקוד שלכם בהערות באנגלית.
11. 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.
12. **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.**
13. השתמשו ב redirection - כדי להפנות את הפלט לקובץ טקסט.
14. השמשו באתר הבדיקה העצמית לבדיקה וקבלת פלט צפוי.
15. אין להדפיס רווחים ותווים שלא התבקשתם להדפיס.
16. ההגשה הינה אלקטרונית דרך אתר הקורס. קובץ ההגשה יהיה מסוג ZIP. לא יתקבלו פורמטים אחרים. קובץ ההגשה יכיל את הקבצים הבאים בלבד ללא 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.
17. **חובה לשמור את אישור ההגשה שמקבלים מהמערכת לאחר שמגישים, עד לסיום הקורס.**
18. יש להקפיד להגיש את כל הקבצים בדיוק עם השמות שמופיעים לעיל. הגשה שלא תעמוד בתנאי זה לא תתקבל ע"י המערכת!
19. **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.**

שאלה 1: משפטי תנאי.

אליס ובוב נפגשים כל ערב לצפות בסדרה האהובה עליהם. בכל צהריים אליס שולחת הודעה לבוב עם המאכל שמתחשק לה לאכול ולשתות בעת הצפייה, אליס לא חוששת לעשות זאת שכן ההתכתבויות בין אליס ובוב מוצפנות. לצערו של בוב תוכנת הפיענוח שלו התקלקלה, כדי שבווב לא יאכזב את אליס נעזר לבוב לכתוב תוכנית פיענוח חדשה.

ההודעה המוצפנת שאליס שולחת מורכבת ממספר לא ידוע של אותיות קטנות באנגלית. (ללא רווחים, מספרים או כל תו אחר). כללי הפיענוח של ההודעה הינם:

1. אם סכום ערכי ה-ASCII של האותיות שנשלחו הינו זוגי, אז אליס רוצה לשתות בירה, אחרת, אליס רוצה לשתות יין.
 2. האותיות הזוגיות הן b, d, f, \dots והאי-זוגיות הן a, c, e, \dots (בהתאמה לזוגיות ערכי ה-ASCII). אם מספר האותיות הזוגיות במילה גדול או שווה למספר האותיות האי-זוגיות אז אליס רוצה לאכול פיצה.
 3. הודעה מונוטונית-לא-יורדת הינה הודעה שהאותיות בה מופיעות בסדר לא-יורד ביחס לסדר הופעת האותיות באלפבית האנגלי ($a < b < c < d < \dots < z$). אם ההודעה שהתקבלה הינה מונוטונית לא-יורדת אז אליס רוצה לאכול ריזוטו.
- הערות:
- הודעה מונוטונית לא-יורדת יכולה להכיל יותר אותיות זוגיות מאי-זוגיות או מספר שווה של אותיות זוגיות ואי-זוגיות, במקרה זה מתקיימים תנאי 2 וגם תנאי 3.
 - דוגמה 1: $abdf$ שמכילה 3 אותיות זוגיות b, d, f ואות אי-זוגית אחת a .
 - דוגמה 2: $abcdef$ שמכילה 3 אותיות זוגיות ו-3 אותיות אי-זוגיות.
- במקרה זה, בו תנאי 2 וגם תנאי 3 מתקיימים אליס רוצה לאכול פיצה ולא ריזוטו. (תנאי 2 גובר על תנאי 3).
4. בכל מקרה אחר, אליס רוצה לאכול פסטה.

מהלך התוכנית:

1. נדפיס למסך: "Hey Bob, Please enter Alice's message:\n". - ניתן לקרוא לפונקציה `AskBobForAliceMessage()` לשם הדפסה זו. (מומשה בקובץ `hw2q2.c`).
2. בוב יקליד את הודעתה של אליס, אך יתכן שבוב בטעות יקליד רווחים (`spaces`) או ירידות שורה (`enter`) לכן בעת הקליטה יש להתעלם מתווים אלו (בבקשה, עברו שוב על התרגולים כדי לבדוק כיצד לעשות זאת, זה מאוד פשוט). מלבד זאת, מובטח לנו שבוב יקליד רק אותיות קטנות באנגלית (ולכן אין צורך לעשות כל בדיקת קלט אחרת).
3. כאשר בוב יסיים להקליד את ההודעה, בוב יקליד את התו `@`.
4. נכתוב למסך: "Alice wants to eat X and drink Y this evening." כאשר X, Y תלויים בחישובי התוכנית שתיארנו לעיל:

○ X יכול להיות: Pizza, Risotto, Pasta.

○ Y יכול להיות: Beer או Wine.

כל האפשרויות להדפסה נתונות כפונקציות `hw2q1.c`.

הערות נוספות:

1. מובטח שבוב יקליד לפחות אות קטנה באנגלית אחת במהלך הקלדת ההודעה (כלומר לא יוקלד רק `@`).
2. מובטח שבוב יקליד `@` בסוף ההודעה.

דוגמת הרצה:

```
Hey Bob, Please enter Alice's message:
    abd

fh
    j\

@
Alice wants to eat Pizza and drink wine this evening.
```

- ההודעה הינה abdfhj\ אך בוב הקליד המון Spaces ו- Enters שיש להתעלם מהם.
- ההודעה מונוטונית.
- מספר האותיות הזוגיות גדול ממספר האותיות האי זוגיות.
- סכום הערכים האסקיים של האותיות אי זוגי.
- לכן לפי הכללים אליס רוצה לאכול פיצה ולשתות יין.

שאלה 2: מערכים.

נכתוב תוכנית הקולטת מהמשתמש/ת מספר בינרי שאורכו 32 ביטים לכל היותר ומבצעת אחת משתי פעולות:

1. Shift Right – הפעולה מזיזה כל ביט במספר שנקלט מיקום אחד ימינה.
 - לדוגמה $\text{ShiftRight}(10110) = 01011$.
 - נבחין שהביט השמאלי ביותר מתאפס.
 - נבחין שהביט הימני ביותר "נמחק".
2. Unsigned int value – הפעולה מחשבת ערכו העשרוני של המספר השלם החיובי המיוצג על ידי המספר הבינרי.
 - נזכיר שבהינתן מספר בינרי $(b_{31}b_{30} \dots b_1b_0)_2$ כאשר $b_i \in \{0,1\}$ לכל $0 \leq i \leq 31$ המספר העשרוני החיובי המיוצג על ידי המספר הבינרי הינו $b_{31} \cdot 2^{31} + b_{30} \cdot 2^{30} + \dots + b_1 \cdot 2^1 + b_0 \cdot 2^0$.

הערה – בדומה לשאלה 1, גם בתוכנית זאת יש להדפיס מספר שורות למסך. מימשנו את פונקציות ההדפסה בקובץ hw2q2.c, נמליץ להשתמש בהן כדי למנוע טעויות הקשורות במחרוזות המודפסות, אך זו אינה חובה.

מהלך התוכנית:

1. נדפיס למשתמש/ת את ההודעה: "Please enter a binary digit:\n" - ניתן לקרוא לפונקציה `PrintEnterMessage` לשם הדפסה זו. (מומשה בקובץ hw2q2.c).
 - 2. נסרוק מהמשתמש/ת את המספר הבינרי. מובטח ש:
 - יוקלדו רק אפסים ואחדים (ללא רווחים וללא ירידות שורה).
 - יוקלד לכל הפחות ביט אחד.
 - יוקלדו לכל היותר 32 ביטים (כלומר ממש פחות מ-33 ביטים).
 - בסיום הקלדת הספרות, יוקלד התו '!' . (מיד עם סיום הקלדת הספרות, ללא רווחים/ירידות שורה בין הספרה האחרונה לבין התו '!').
3. נדפיס למשתמש/ת את ההודעה: "Scanned binary number is:\n" – ניתן לקרוא לפונקציה `PrintScannedBinaryNumbers()` לשם הדפסה זו. (מומשה בקובץ hw2q2.c).
4. נדפיס למשתמש/ת את המספר שנקלט. ההדפסה צריכה לכלול 32 ביטים, במקרה בו נסרק מספר בינרי בעל פחות מ-32 ביטים, נדפיס אפסים משמאל למספר שנקלט (ראו דוגמת הרצה).
5. נדפיס למשתמש/ת הודעת בחירה בין הפעולות:
"Enter S for shift-right or D of unsigned int representation:\n"
`PrintOperationChoiceMessage()` לשם הדפסה זו. (מומשה בקובץ hw2q2.c).
 - 6. נסרוק מהמשתמש/ת את בחירתו. מובטח ש:
 - יוקלד S או D ולאחר מכן Enter לסיום הקלט. (לא יוקלד כל תו אחר).
7. אם הוקלד S:
 - נדפיס: "Shifted binary number is:\n" . ניתן לקרוא לפונקציה `PrintShiftedBinaryNumbers()` לשם הדפסה זו. (מומש בקובץ hw2q2.c).
 - נדפיס את כל 32 הביטים של המספר הבינרי לאחר ביצוע ה-`shift right`. (ראו הערה קודמת למקרה בו נסרק מספר בן פחות מ-32 ביטים וכן את דוגמת ההרצה).
8. אם הוקלד D:
 - נדפיס: "Unsigned int representation is:\n" . ניתן לקרוא לפונקציה `PrintUnsignedIntRepresentation()` לשם הדפסה זו. (מומש בקובץ hw2q2.c)
 - נדפיס את הערך העשרוני השלם האי-שלילי של המספר הבינרי.

דוגמת הרצה לפעולת Right Shift :

```
Please enter a binary digit:
1011101!
Scanned binary number is:
00000000000000000000000001011101
Enter S for shift-right or D of unsigned int representation:
S
Shifted binary number is:
0000000000000000000000000101110
```

- נבחין שההדפסות כוללות 32 ביטים (אפסים משמאל) בעוד שהמספר שהוקלד 1011101 בן פחות מ-32 ספרות בינריות.

דוגמת הרצה לפעולת Unsigned int value :

```
Please enter a binary digit:
11111111111111111111111111111111!
Scanned binary number is:
11111111111111111111111111111111
Enter S for shift-right or D of unsigned int representation:
D
Unsigned int representation is:
4294967295
```

שאלה 3: מערכים דו-מימדיים.

נכתוב תוכנית הקולטת מהמשתמש מטריצה ריבועית (מספר השורות שווה למספר העמודות) עם כניסות מטיפוס int מגודל לכל היותר 25×25 ולכל הפחות 2×2 ובודקת האם המטריצה היא נילפוטנטית מאינדקס $k \geq 1$.

הגדרה: מטריצה ריבועית M הינה נילפוטנטית מאינדקס $k \geq 1$ אם $M^k = 0$ (כאשר $M^k := \underbrace{M \cdot M \cdot \dots \cdot M}_{\text{multiply the matrix by itself } k \text{ times}}$)

היא מטריצת ה-0, מטריצה שכל כניסותיה 0) וגם לכל $i < k : M^i \neq 0$. במילים אחרות, k הוא המספר הטבעי הקטן ביותר כך שאם מכפילים את המטריצה בעצמה k פעמים אז מתקבלת מטריצת האפס.

- נעיר שיש מטריצות עבורן $k = \infty$, לדוגמה מטריצת היחידה.
- נעיר שמטריצת האפס (מטריצה שכל כניסותיה אפסים) הינה נילפוטנטית מאינדקס 1.
- המטריצה: $\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$ הינה נילפוטנטית מאינדקס 2.
- המטריצה: $\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$ הינה נילפוטנטית מאינדקס 3.

מהלך התוכנית:

1. נדפיס למשתמש/ת: "Please enter matrix dimension: " - ניתן לקרוא לפונקציה PrintEnterMatrixDimension לשם הדפסה זו. (מומשה בקובץ hw2q3.c).
2. נקלוט מהמשתמש/ת את המימד של המטריצה אותה התוכנית תבדוק, נזכיר שהמימד הוא לפחות 2 ולכל היותר 25.
 - מובטח שיוקלד ערך בתחום (אין צורך לבדוק את הקלט).
3. נדפיס למשתמש/ת: "n:\nPlease enter your matrix: " ניתן לקרוא לפונקציה PrintEnterMatrix לשם הדפסה זו. (מומשה בקובץ hw2q3.c).
4. נסרוק מהמשתמש/ת את המטריצה.
 - מובטח שמספר הערכים שיקלטו יהיה כמספר הכניסות במטריצה.
 - מובטח שהערכים שיוקלדו יהיו מספרים שלמים בלבד.
5. נדפיס למשתמש/ת: "n:\nScanned matrix is: " ניתן לקרוא לפונקציה PrintScannedMatrixIs לשם הדפסה זו. (מומשה בקובץ hw2q3.c).
6. נדפיס למסך את המטריצה שנקלטה, ההדפסה צריכה להתבצע שורה אחר שורה, עם רווח לאחר כל איבר שמודפס (כולל האיבר האחרון בשורה). ראו דוגמת הרצה.
7. נדפיס למשתמש/ת: "Please enter nilpotency index: " ניתן לקרוא לפונקציה PrintEnterNilpotencyIndex לשם הדפסה זו. (מומשה בקובץ hw2q3.c).
8. נסרוק מהמשתמש את אינדקס הנילפוטנטיות.
 - מובטח שיוקלד ערך שלם $1 \leq k$ בטווח היצוג של int.
9. נבדוק האם המטריצה שנסרקה נילפוטנטית מאינדקס k .
 - אם כן, נדפיס: "n:\nIt is %d-nilpotent matrix!". יש להעביר להדפסה את אינדקס הנילפוטנטיות שנקלט מהמשתמש/ת בשלב 8. ניתן לקרוא לפונקציה PrintItIsKNilpotentMatrix(int k) המקבלת את k אינדקס הנילפוטנטיות שסרקנו בשלב 8 ומבצעת את ההדפסה הנ"ל. (מומש בקובץ hw2q3.c).
 - אם לא, נדפיס: "n:\nIt is not %d-nilpotent matrix!". יש להעביר להדפסה את אינדקס הנילפוטנטיות שנקלט מהמשתמש בשלב 8. ניתן לקרוא לפונקציה PrintItIsNotKNilpotentMatrix(int k)

המקבלת את k אינדקס הנילפוטנטיות שסרקנו בשלב 8 ומבצעת את ההדפסה הנ"ל. (מומש בקובץ hw2q3.c).

הערה – הגדרת כפל מטריצות: בהינתן A, B צמד מטריצות ריבועיות מסדר $n \times n$ האיבר (i, j) במכפלת המטריצות AB הינה: $AB_{(i,j)} = Row_A(i) \odot Col_B(j) = \sum_{k=1}^n A_{(i,k)} \cdot B_{(k,j)}$. כאשר \odot מסמן את המכפלה הסקלרית בין וקטורים. לדוגמה עבור המטריצות: $A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}$ מתקיים:

$$AB = \begin{pmatrix} (a_{11}b_{11} + a_{12}b_{21}), (a_{11}b_{12} + a_{12}b_{22}) \\ (a_{21}b_{11} + a_{22}b_{21}), (a_{21}b_{12} + a_{22}b_{22}) \end{pmatrix}$$

דוגמאת הרצה:

```
Please enter matrix dimension: 3
Please enter your matrix:
2 2 -2
5 1 -3
1 5 -3
Scanned matrix is:
2 2 -2
5 1 -3
1 5 -3
Please enter Nilpotency index: 3
It is 3-nilpotent matrix!
```