

## תרגיל בית 5

מועד אחרון להגשה: יום ה' – 23:59 30.06.2022

המתרגל האחראי על תרגיל זה: רואי בנימין

פורום לשאלות שאלות: [פיאצה](#)

### הנחיות:

- הגשה ב**בודדים**. עליכם לכתוב את הפתרונות לבד ולהגיש ביחידים.
- קראו את השאלות בעיון לפני שתתחילו בפתרון.
- 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 עותק של האישור המראה שהייתם במילואים (טופס 3010). אם האישור יגיע אליכם בתאריך מאוחר, יש להודיע על כך למתרגל האחראי לפני תאריך הגשת התרגיל.
- לא ניתן לערער על תוצאות הבדיקה האוטומטית.
- **שימו לב! הבדיקה בתרגיל זה הינה אוטומטית, ולכן הקפידו להדפיס בדיוק בפורמט שהתבקשתם ובידקו עם אתר הבדיקה ועם DiffMerge את הפלט שלכם מול הפלט של הדוגמאות שקיבלתם.**
- בתרגיל זה מותר להשתמש בפונקציות שנלמדו בתרגולים מהספריות `stdlib.h`, `stdbool.h`, `stdio.h` ו-`string.h`, למעט במקרים בהם נאמר אחרת. החומר הנדרש לתרגיל זה הוא כל חומר הקורס.
- ההגשה הינה אלקטרונית וב**בודדים** 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 `hw5q1.c` עבור שאלה 1.
  - קובץ פתרון `hw5q2.c` עבור שאלה 2.
- חובה לשמור את קוד 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.
- במהלך התרגיל נממש פונקציות חסרות בקבצים `hw5q1.c` ו-`hw5q2.c`. שימו לב, אין לשנות את הקוד הקיים בקבצים הללו, אך מותר להוסיף קבועים ופונקציות עזר.
- מומלץ לקרוא את כל הקוד הקיים, גם את החלקים אשר לא מימשתם, ולוודא כי אתם מבינים אותו. הבדיקה הסופית תיעשה ע"י הרצת הקוד המלא.
- לאורך כל התרגיל, כאשר נציין אינדקס של איבר, שורה או עמודה במערך חד-ממדי או דו-ממדי, הכוונה לאינדקסים המתחילים ב-0, כפי שמקובל בשפת C.

## שאלה 1 – Backtracking

ניתן להניח בכל סעיפי שאלה זו שהקלט תקין.

### סעיף א'

כתבו פונקציה שחתימתה:

```
int ShortestFullPath(int roads[N][N]);
```

הפונקציה מקבלת מטריצת כבישים roads בגודל  $N \times N$  (בתרגיל זה  $N = 4$ ).

המטריצה roads מכילה מספרים שלמים על פי הכלל הבא:

האיבר  $(i, j)$  שווה לאורך הכביש עליו ניתן לנסוע מהבית  $i$  לבית  $j$  (אך לא להיפך) במידה ויש כזה, ושווה ל-NO\_ROAD (קבוע המוגדר כ-1) במידה ואין כזה. אלכסון המטריצה מכיל אפסים (כי בית רחוק מעצמו מרחק 0). **המטריצה לא בהכרח סימטרית.**

הפונקציה מחזירה את אורך המסלול הקצר ביותר שמתחיל בבית מספר 0, מסתיים בבית מספר 0 ועובר בכל שאר הבתים בדיוק פעם אחת. אם אין מסלול כזה הפונקציה מחזירה NO\_PATH (קבוע המוגדר כ-1). נקרא למסלול זה "המסלול האופטימלי".

### סעיף ב'

כתבו את פונקציית ה-main שקולטת מהמשתמש את המטריצה roads שורה אחר שורה, ומדפיסה למסך את אורך המסלול האופטימלי.

### קבועים נתונים

הגדרות הקבועים הנתונים לכם:

```
#define N 4
```

```
#define NO_ROAD -1
```

```
#define NO_PATH -1
```

### פורמט ההדפסה

ראשית תודפס הודעה המבקשת להכניס את ערכי המטריצה בעזרת הפונקציה:

```
void PrintRoadsInputMessage();
```

המשתמש יכניס את ערכי המטריצה roads שורה אחרי שורה.

לאחר מכן הדפסת אורך המסלול האופטימלי למסך תתבצע בעזרת הפונקציה:

```
void PrintLenOfShortestFullPath(int min_len);
```

המקבלת את אורך המסלול האופטימלי min\_len.

### דוגמאות הרצה

```
Please enter the roads 4X4 matrix row by row:
```

```
0 2 -1 2
```

```
-1 0 2 -1
```

```
1 -1 0 2
```

```
2 1 -1 0
```

```
The shortest path is of length: 6
```

```
Please enter the roads 4X4 matrix row by row:
```

```
0 1 -1 3
```

```
1 0 1 -1
```

```
-1 1 0 1
```

```
2 -1 1 0
```

```
The shortest path is of length: 5
```

## שאלה 2 – רקורסיה ומיון

ניתן להניח בכל סעיפי שאלה זו שהקלט תקין. בשאלה זו ניתן להניח שכל הקלטים הם מספרים אי-שליליים וניתנים לייצוג ע"י unsigned int. בנוסף, מותר לכם להגדיר פונקציית מעטפת שתקרא לפונקציה הרקורסיבית, ושהפונקציה הרקורסיבית תקבל יותר פרמטרים ממה שמוגדר בשאלה.

### סעיף א'

פלינדרום הוא מספר שלם אי-שלילי שניתן לקריאה משני הכיוונים כך שערכו לא ישתנה. לדוגמה, המספרים 1, 45654, 123321 הם פלינדרומים, בעוד המספרים 12312, 13 אינם פלינדרומים. ממשו פונקציה רקורסיבית, המקבלת מספר שלם אי-שלילי (unsigned int), ומחזירה true במקרה שהמספר פלינדרום, ואחרת מחזירה false. חתימת הפונקציה -

```
bool isPalindrome(unsigned int x);
```

רמז: על מנת לפשט את המימוש, שקלו לייצר מערך של ספרות מהמספר שאתם מקבלים, ולעבוד איתו. ניתן להסתמך על העובדה שהמספר המקסימלי שניתן לייצג ע"י unsigned int מכיל 10 ספרות.

### סעיף ב'

נגדיר ספרה מייצגת של מספר אי-שלילי שלם כמכפלה של כל ספרות המספר אם מכפלה זו חד ספרתית. אם המכפלה אינה חד-ספרתית, כופלים את הספרות של המכפלה וכך הלאה עד שמגיעים למספר חד-ספרתי והוא יהיה הספרה המייצגת של המספר המקורי. לדוגמה, הספרה המייצגת של המספר 13242 מחושבת כך:

$$48 = 1 * 3 * 2 * 4 * 2$$

$$32 = 4 * 8$$

$$6 = 3 * 2$$

לכן הספרה המייצגת היא 6.

ממשו פונקציה רקורסיבית, המקבלת מספר שלם אי-שלילי (unsigned int), ומחזירה את הספרה המייצגת שלו. חתימת הפונקציה -

```
int represent(unsigned int x);
```

### סעיף ג'

כתבו את פונקציית main המקבלת מהמשתמש מספר n ולאחר מכן המשתמש יכניס n מספרים שלמים אי-שליליים. התכנית תמייין את המספרים, ותדפיס אותם אחד בכל שורה לפי הסדר הבא:

מספר שהוא פלינדרום יהיה לפני מספר שאינו פלינדרום. אם שני המספרים פלינדרומים או ששניהם אינם פלינדרומים, המספרים ימוינו לפי הספרה המייצגת שלהם (בסדר עולה). אם גם הספרה המייצגת שלהם זהה, המספרים ימוינו מהקטן לגדול.

```
5
11
10
344
15651
87
```

Sample input:

```
15651
11
10
87
344
```

Sample output:



**להתחיל את  
תרגיל הבית  
מוקדם**



**לדחות לרגע  
האחרון  
ולעשות לילה  
לבן**

בהצלחה!