234114-7 מבוא למדעי המחשב, סמסטר חורף 2018-2019

תרגיל בית 5

מועד אחרון להגשה: **יום ד' – 23.01.2019 23:59**

המתרגל האחראי על תרגיל זה: **יאיר ריעאני**

משרד: טאוב 420
E-mail: syairr@campus.technion.ac.il

הנחיות:

- הגשה ב**בודדים**. עליכם לכתוב את הפתרונות לבד ולהגיש ביחידים.
- קראו את השאלות בעיון לפני שתתחילו בפתרונן.
- 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 על התרגיל לפני תאריך הגשת התרגיל.
- **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.**
- בתרגיל זה מותר להשתמש בפונקציות שנלמדו בתרגולים מהספריות stdio.h, stdbool.h, stdlib.h ו- string.h, למעט במקרים בהם נאמר אחרת. החומר הנדרש לתרגיל זה הוא כל חומר הקורס.
- ההגשה הינה אלקטרונית ו**בבודדים** 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. בנוסף, אין להדפיס שום פלט במהלך הפונקציות אותן אתם ממשים.
- 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- מימדי או דו- מימדי, 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 NxN (בתרגיל זה N = 4).

המטריצה roads מכילה מספרים שלמים על פי הכלל הבא:

האיבר ה- (i,j) שווה לאורך הכביש עליו ניתן לנסוע מהבית ה- i לבית ה- 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.

<u>דוגמת הרצה</u>

```
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
```

שאלה 2 – מיון

ניתן להניח בכל סעיפי שאלה זו שהקלט תקין.

בשאלה זו נמיין מחרוזות על פי כללי השוואה שונים.

סעיף א'

כתבו פונקציה שחתימתה:

```
int CompareStrings(char* str1, char* str2, int rule);
```

הפונקציה מקבלת שתי מחרוזות str1 ו- str2 ומס' שלם rule.

הפונקציה מחזירה מספר חיובי כלשהו אם str1 גדולה מ- str2, מחזירה מספר שלילי כלשהו אם str2 גדולה מ- str1, ומחזירה 0 אם str1 שווה ל- str2.

המשמעות של "גדולה מ-" ושל "שווה ל-" משתנה בהתאם לערך rule, שמקבל אחד משני ערכים: LEX ו- ABC_APPEAR המוגדרים בעזרת define.

<u>עבור rule = LEX</u>

עבור שתי מחרוזות s1 ו- s2 נאמר ש- s1 גדולה מ- s2 במידה והיא גדולה ממנה על פי הסדר הלקסיקוגרפי. **ההשוואה לא מבדילה בין אותיות קטנות לגדולות.** לגבי תווים שאינם אותיות אנגליות, תו בעל ערך ASCII גדול יותר, ייחשב מתקדם יותר.

דוגמאות:

1. המחרוזת "AB" גדולה מהמחרוזת "AA", וכן "AAA" גדולה מ- "AA".
2. "aa" ו- "AA" שוות.
3. "?" גדולה מ- "!" מכיוון שערך ה- ASCII של סימן שאלה גדול מערכו של סימן קריאה.

<u>עבור rule = LET_DIVER (קיצור של letters diversity)</u>

עבור שתי מחרוזות s1 ו- s2 נאמר ש- s1 גדולה מ- s2 <u>במידה וכמות סוגי האותיות מהאלפבית האנגלי</u> הנמצאת ב- s1 גדולה מהכמות הנמצאת ב- s2. **ההשוואה לא מבדילה בין אותיות קטנות לגדולות.**

דוגמאות:

1. המחרוזת "ab" גדולה מהמחרוזת "CCCccccCC", מכיוון שבמחרוזת הראשונה מופיעות שתי אותיות (a ו- b) מהאלפבית האנגלי, ובמחרוזת השנייה מופיעה בסך הכל אות אחת (c).
2. "aaabbbb" ו- "CCCddd" שוות.

סעיף ב'

ממשו את הפונקציה:

```
void SortString(char* str_arr[], int n, int rule);
```

המקבלת מערך מחרוזות str_arr באורך n, וממיינת אותו בהתאם לכלל ההשוואה rule בין המחרוזות. כלומר, אם מחרוזת s1 גדולה ממחרוזת s2, כפי שמוגדר על פי rule, היא צריכה להופיע אחריה במערך הממויין.

בנוסף, המיון צריך להיות יציב. כלומר, בהינתן שתי מחרוזות שוות s1 ו- s2 המקיימות שבמערך המקורי s1 ממוקמת אחרי s2, מתקיים שגם במערך הממויין s1 ממוקמת אחרי s2.

דוגמאות מובאות בסעיף הבא.

סעיף ג'

כתבו את פונקציית ה- main שקולטת מהמשתמש את מערך המחרוזות ומדפיסה למסך את המחרוזות באופן ממויין.

קבועים נתונים

הגדרות הקבועים הנתונים לכם:

```
#define LEX 1

#define LET_DIVER 2

#define MAX_LEN 20
```

הקבוע האחרון מגדיר את אורך המחרוזת המקסימלי.

פורמט ההדפסה:

ראשית תודפס למסך הודעה המבקשת מהמשתמש להכניס את מספר המחרוזות n בעזרת הפונקציה הנתונה:

```
void PrintNumStringsInputMessage();
```

שלאחריה ייקלט מספר שלם n מהמשתמש.

לאחר מכן תודפס הודעה למשתמש המבקשת ממנו להכניס n מחרוזות בעזרת הפונקציה:

```
void PrintStringsInputMessage(int n);
```

שלאחריה תיקלטנה n מחרוזות.

ניתן להניח שאורך כל מחרוזת לא עולה על MAX_LEN = 20 תווים.

לאחר מכן תודפס הודעה המבקשת מהמשתמש להכניס את כלל ההשוואה בעזרת הפונקציה:

void PrintRuleOfComparisonInputMessage();

לאחר מכן יודפס מערך המחרוזות הממויין למסך בעזרת הפונקציה:

void PrintSortedStrings(char* str_arr[], int n);

שמקבלת מערך מחרוזות str_arr באורך n, ומדפיסה אותו על פי הסדר.

במידה שבמהלך התכנית ישנה שגיאה בהקצאת זיכרון, תודפס הודעה למסך באמצעות פונקציית ההדפסה:

void PrintAllocationError();

והתכנית תסתיים.

<u>דוגמאות הרצה</u>

```
Please enter the number of strings:
3
Please enter the 3 strings:
bc
ab
AB
Please enter the rule of comparison between strings.
1: Lexicographic order.
2: By the diversity of letters.
1
The sorted strings are:
ab
AB
Bc
```

```
Please enter the number of strings:
5
Please enter the 5 strings:
abcd
adf
nmmmmmmmMMMMMM
gh
11kk%$#
Please enter the rule of comparison between strings.
1: Lexicographic order.
2: By the diversity of letters.
2
The sorted strings are:
11kk%$#
nmmmmmmmMMMMMM
gh
adf
abcd
```

בהצלחה!