הנחיות כלליות

יש לשלוח את הקבצים באמצעות מערכת ההגשה לפני חלוף התאריך 7/12/19.

ניתן להגיש את התרגיל באיחור עם קנס אוטומטי על פי הפירוט הבא:

- יום איחור - קנס של 5 נקודות (ציון מקסימלי – 95).
- יומיים איחור - קנס של 15 נקודות (ציון מקסימלי – 85).
- שלושה ימי איחור - קנס של 30 נקודות (ציון מקסימלי – 70).

לאחר מכן לא יהיה ניתן להגיש את התרגיל (ציון 0).

המתרגל האחראי על התרגיל הוא אייל.

שאלות בנוגע לתרגיל יש לפרסם **באופן ציבורי בפורום הקורס** בלבד! רק אם לא התקבלה תשובה לאחר 24 שעות, יש לשלוח מייל לכתובת eyal.dayan@biu.ac.il עם קישור לדיון הרלוונטי.

בקשות להארכה מסיבות מוצדקות (מילואים, לידה וכו') יש לפרסם **באופן פרטי בפורום הקורס** בלבד (יש למען את הפוסט ל-instructors). בכל בקשה יש לציין שם מלא, שם משתמש במערכת ההגשה, תעודת זהות והאם אתם ממדעי המחשב או מתמטיקה.

יש להקפיד מאוד על הוראות עיצוב הקלט והפלט, בדיוק על פי הדוגמאות המצורפות.
שימו לב להנחיות במסמך ה-Coding Style המפורסם באתר הקורס.

עליכם לכתוב קוד על פי ההנחיות ולוודא שקיבלתם 100 בבדיקה האוטומטית הראשונית, וכן שהתרגיל מתקמפל ורץ על שרתי המחלקה (u2) ללא שגיאות או אזהרות.
תרגיל שלא עומד בסטנדרטים הבסיסיים הללו יגרור **ירידה משמעותית בציון התרגיל**, בשל הטרחה שהוא מייצר בתהליך הבדיקה שלו.

להזכירכם העבודה היא אישית. "עבודה משותפת" דינה כהעתקה. התרגיל נבדק על ידי מערכת ההגשה האוטומטית גם מהבחינה הזו, ותרגיל שהועתק יגרור ציון 0 **לכל הגורמים** השותפים בהעתקה. אתם יכולים לדון בגישות לפתרון התרגיל באופן תיאורטי, אך אין לשתף קוד בשום צורה.

בפיתוח הקוד ניתן להשתמש בכל סביבת עבודה, העיקר הוא שתדעו איך לקחת את קבצי הקוד מתוך הסביבה הזו, לבדוק אותם על שרתי האוניברסיטה ולהגיש אותם באמצעות מערכת ההגשה.
דוגמאות לחלק מהסביבות האפשריות:

IDEs (Integrated Development Environment):

- Visual studio
- Clion
- Eclipse
- Xcode

Text Editors:

- Atom
- Sublime
- Notepad++
- Vim

בהצלחה!

תרגיל 3 – ass3

בתרגיל זה עליכם לממש פונקציות בקובץ יחיד בשם ass3.c

משקל התרגיל מתוך ציון התרגול: 10%.

- לתרגיל זה מצורף קובץ בשם ass3.h המכיל הצהרות, וקובץ בשם main.c המכיל פונקציית main ופונקציות נוספות. אין להגדיר פונקציות בשמות האלו בקובץ ass3.c שאתם מגישים. בנוסף, אין צורך להגיש את הקבצים המצורפים.
- בתרגיל זה עליכם להשתמש בקבוע define חיצוני בשם MAX.
- בתרגיל זה עליכם להגדיר משתנה גלובלי יחיד ולהשתמש בו בהתאם להנחיות התרגיל בלבד.
- בתרגיל זה מותר להשתמש בפונקציה log10 מתוך הספריה math.h ולהשתמש בה בהתאם להנחיות התרגיל בלבד. מעבר לכך אין להשתמש בפונקציות מתוך הספריה. יש להוסיף לפקודת הקימפול את הדגל lm- לצורך קישור הספריה.

פקודת הידור לדוגמה (כי ערכו של MAX יכול להיות אחר) עבור התרגיל:

```
gcc ass3.c main.c -std=c99 -lm -DMAX=1000
```

מצורף הפלט עבור הקוד המופיע בקובץ main.c (כאשר הפונקציות ממומשות), עם הפקודה הזו.
אתם יכולים לסמן בהערה את הפקודות בפונקציית ה-main אם תרצו לבדוק את הפונקציות בנפרד.

```
iter: 1.5^3 = 3.375
rec : 1.5^3 = 3.375
effi: 1.5^3 = 3.375

00001 00002 00003 00004
00555 00006 00077 00008
09999 00010 11111 00012

Yes!
No!
Yes!
```

שימו לב שמסמך ה-Coding Style עודכן. כמו כן, אם הורדתם את מצגות התרגול, מומלץ לוודא שיש ברשותכם את הגרסה העדכנית (בדרך כלל אני מעלה אותה בסוף השבוע מחדש).

בנוסף, אני מזכיר שאת כל הפניות בנוגע לתרגיל יש לבצע באמצעות הפיאצה. יש לשלוח לי מייל רק אם פספסתי פוסט שלכם בטעות, ולא קיבלתם מענה במשך 24 שעות (ואתם רואים שפוסטים חדשים יותר מקבלים מענה).

חלק ראשון – חישוב חזקה

להלן שלוש הגדרות עבור חישוב הביטוי x^y:

הגדרה איטרטיבית	$x^y = x \cdot x \cdot \ldots \cdot x$	$x^y = \prod_{i=1}^{y} x$
הגדרה רקורסיבית נאיבית	$x^y = x \cdot x^{y-1}$	
הגדרה רקורסיבית יעילה	$x^y = x^{y/2} \cdot x^{y/2}$	

בחלק זה עליכם לממש שלוש פונקציות:

```
double iterPow(double x, int y);
double recPow(double x, int y);
double recEffiPow(double x, int y);
```

כל אחת מהפונקציות מקבלת מספר עשרוני x ומספר שלם y ומחזירה את ערכו של x בחזקת y.

- בפונקציה iterPow יש לחשב את החזקה על פי ההגדרה האיטרטיבית.
- בפונקציה recPow יש לחשב את החזקה על פי ההגדרה הרקורסיבית הנאיבית.
- בפונקציה recEffiPow יש לחשב את החזקה על פי ההגדרה הרקורסיבית היעילה.

מספר דגשים בנוגע להגדרות:

- שימו לב שההגדרות הן מתמטיות, ייתכן שיידרשו התאמות קלות בשל התכונות של שפת c.
- כל מספר בחזקת 0 הוא 1 (כולל 0).
- כל מספר בחזקת 1 נשאר ללא שינוי.
- 0 בחזקת מספר שלילי הוא אינסוף (אפשר לשמור זאת במשתנה מטיפוס עשרוני).

הנחיות נוספות:

- מותר להגדיר פונקציות עזר.
- בפונקציה האיטרטיבית (ובפונקציות העזר שלה) אין להשתמש ברקורסיה.
- בפונקציות הרקורסיביות (ובפונקציות העזר שלהן) אין להשתמש בלולאות.

נרצה להשוות את היעילות של המימושים השונים. לכן, יש להגדיר משתנה גלובלי שלם בשם counter ולהשתמש בפקודה counter++ פעם אחת בשורה הראשונה של כל פונקציה רקורסיבית (וכל פונקציית עזר שלה), ובתוך כל לולאה בפונקציה האיטרטיבית (ובכל לולאה הנמצאת בכל פונקציית עזר שלה). אם הגדרתם לולאות מקוננות - פקודה הכתובה בתוך לולאה פנימית נחשבת גם עבור כל לולאה המכילה אותה, לכן מספיק לכתוב אותה רק בלולאה הפנימית.

בבדיקה הידנית של התרגיל נתייחס ליעילות הזו (מספר הקריאות לפונקציות ברקורסיה, ומספר האיטרציות בלולאה).

אתם יכולים להדפיס את המונה בנקודות שמופיעות בקובץ main.c כדי לבדוק את עצמכם בהיבט הזה.

חלק שני – מערכים

בחלק זה עליכם לממש שתי פונקציות:

```
int isPermutation(int arr1[], int size1, int arr2[], int size2);
void printArr2D(int arr[][MAX], int size1, int size2);
```

- בפונקציה isPermutation עליכם לקבל שני מערכים (ואת מימדיהם), ולבדוק האם אחד המערכים מהווה פרמוטציה (סידור מחדש) של השני. אם כן – יש להחזיר 1, אחרת יש להחזיר 0.
- בפונקציה printArr2D עליכם לקבל מערך דו-מימדי של מספרים שלמים, ולהדפיס אותו כטבלה ישרה – כלומר יש לרפד באפסים את המספרים כך שלכולם יהיה את אותו מספר ספרות שיש למספר הארוך ביותר (ואותו אין לרפד באפסים כלל).

ניתן להניח את ההנחות הבאות על הקלט:

- הערכים בתוך המערכים המועברים לפונקציה isPermutation יהיו בטווח [0 : MAX-1].
- ערכם של המימדים המועברים לפונקציה printArr2D לא יהיו גדולים מערכו של MAX.
- הערכים בתוך המערך המועבר לפונקציה printArr2D לא יהיו שליליים.

הנחיות נוספות:

-
- בפונקציה printArr2D **בלבד**, מותר להשתמש בפונקציה log10 מתוך הספריה math.h כדי לחשב את הריווח הנכון. יש להוסיף לפקודת הקימפול את הדגל lm- על מנת להורות ללינקר לקשר את הספריה.
- מי שמעוניין להגדיר פונקציית עזר – שימו לב שניתן להעביר שורה של מערך דו-מימדי לפונקציה כמערך חד-מימדי בעזרת גישה עם אינדקס השורה בלבד:

```
#define ROWS …
#define COLUMNS …
void f(int arr[], int size) {
    ...
}

void main() {
    int arr[ROWS][COLUMNS];
    int i = …
    f(arr[i], COLUMNS);
}
```