

# מבוא לתכנות מערכות 234124

## תרגיל בית 2

### סמסטר אביב 2020

תאריך פרסום: 12/05

תאריך הגשה: 02/06

משקל התרגיל: 5% מהציון הסופי (תקף)

מתרגלים אחראיים: תום הירשברג, אלון פאר

#### 1. הערות כלליות

- שימו לב: לא יינתנו דחיות למועד התרגיל. תכננו את הזמן בהתאם.
- תרגיל זה הינו תרגיל רטוב בלבד, המכיל שני חלקים.
- לשאלות בנוגע להבנת התרגיל יש לשאול בפורום של התרגיל במודל או להגיע לסדנאות. לפני שליחת שאלה - נא וודאו שהיא לא נענתה כבר ב-F.A.Q או במודל, ושהתשובה אינה ברורה ממסמך זה, מהדוגמא ומהבדיקות שפורסמו עם התרגיל.
- קראו מסמך זה עד סופו לפני שאתם מתחילים לממש. יתכן שתצטרכו להתאים את המימוש שלכם לחלק עתידי בתרגיל. תכננו את המימוש שלכם לפני שאתם ניגשים לעבוד. רצוי לעבור על הדוגמא שפורסמה לפני תחילת הפתרון.
- כל חומר הנלווה לתרגיל נמצא על השרת בתיקיה `~mtm/public/1920b/ex2`
- חובה להתעדכן בהודעות באתר הקורס ובעמוד ה-F.A.Q של התרגיל – הכתוב שם מחייב.
- העתקות קוד בין סטודנטים תטופלנה בחומרה!
- מומלץ מאוד מאוד לכתוב את הקוד בחלקים קטנים, להעביר כל חלק בנפרד אל השרת ולבדוק שהוא עובד באמצעות שימוש בטסטים.

#### 2. התקנת פייתון וסביבת העבודה

באתר הקורס, תחת Course Material, במדריך להתקנה ושימוש ב-VSCode מפורטים התקנת פייתון על המחשב האישי ושימוש בסביבת העבודה ובשרת בשפה זו. עקבו אחר ההוראות בסעיף (5) להתקנת פייתון ושימוש בו בסביבת העבודה ובשרת לפני התחלת התרגיל.

## הערות:

- אתם מוזמנים להשתמש בכל סביבה הנוחה לכם, ואפילו עורך טקסט עשוי להספיק עבור תרגיל זה. המלצתנו היא להשתמש בVSCode בשל היתרונות שלו לעבודה משותפת ולתמיכה רחבה בשפת פייתון.
- הגשתכם תיבדק על השרת CSL3 בגרסת פייתון 3.6.3 אשר מותקנת על השרת. למען הסר ספק, אתם יכולים לממש את התרגיל על מחשבכם האישי בכל גרסת פייתון 3.6.3 ומעלה, אך דאגו לבדוק את נכונות מימושכם על השרת לפני ההגשה.

## 3. חלק א – שליפת נתונים מקובץ טקסט באמצעות Python

### 3.1. הקדמה

לקראת אולימפיאדת 2020 שנדחתה וצפויה להתקיים בקיץ של השנה הבאה בטוקיו, נבחרו הסטודנטים של קורס מת"ם בטכניון לדרג את המדינות לפי הישגי הספורטאים מכל מדינה. הישגי מדינה נמדדים במספר וסוגי המדליות שהשיגו נציגיהם במשחקים האולימפיים.

חוקי האולימפיאדה מפורטים להלן:

- כל מתחרה רשאי להשתתף בתחרות אחת או יותר.
- מתחרה זוכה במדליה אם הוא הגיע לאחד משלושת המקומות הראשונים. מקום ראשון זוכה במדליית זהב, מקום שני במדליית כסף ושלישי במדליית ארד. מקומות אחרים אינם זוכים במדליה.
- כל מתחרה מורשה להשתתף בתחרות מסוימת פעם אחת לכל היותר. במידה והוא משתתף בתחרות מסוימת יותר מפעם אחת אז הוא נפסל בתחרות, אך יכול להתחרות בתחרויות אחרות.
- קיימים שלושה סוגים של תחרויות:

1. timed – המנצח בתחרות הוא המתחרה שהשיג את התוצאה הנמוכה ביותר. למשל, בתחרות שחייה המתחרה שהשיג את התוצאה בעלת הזמן המועט ביותר לסיום המקצה הוא המנצח.

2. untimed – המנצח בתחרות הוא המתחרה שהשיג את התוצאה הגבוהה ביותר. למשל, בתחרות קפיצה לגובה המתחרה שהצליח לעבור את הגובה הגבוה ביותר הוא המנצח.

3. knockout – תחרויות מסוג זה הן תחרויות טורניר, כמו למשל כדורגל או כדורסל, בהן התוצאה היא דירוג בטורניר (מקום). מתחרה (competitor) בסוג תחרות כזה נחשב כקבוצה שלמה המתחרה עבור המדינה אותה מייצגת. לכל תחרות מסוג זה מקומות המתחרים נתונים באופן רציף, לדוגמא: בתחרות מסוג basketball, אם נבחרת ישראל (שמוצגת ע"י

competitor\_id אחד לכל הקבוצה) מדורגת במקום 5, אז מובטח כי יהיו נתונות קבוצות שדורגו במקומות 1 עד 4 (אך לאו דווקא לפי סדר מסוים).

- תיקו אינו אפשרי באולימפיאדה, שכן מכשירי המדידה מדויקים ביותר הודות לטכנולוגיה המרשימה והחדשנית בה משתמשים כיום.

### 3.2. תיאור קובץ הקלט

נתון קובץ טקסט בעל מבנה המורכב ממספר סוגי שורות:

```
competitor <competitor id> <competitor country>
competition <competition name> <competitor id> <competition type> <result>
```

כלומר, הקובץ מכיל את שני סוגי השורות הבאים:

1. שורות המתחילות במילה competitor, אשר מתארות מתחרה באולימפיאדה.
2. שורות המתחילות במילה competition, אשר מתארות תוצאה של מתחרה בתחרות.

פירוש הפרמטרים בשני סוגי השורות הוא:

- competitor id – מזהה עבור מתחרה. מספר שלם.
- competitor country – שם מדינה. מחרוזת.
- competition name – שם תחרות. מחרוזת.
- competition type – סוג תחרות, כפי שפורט בסעיף 3.1. מחרוזת.
- result – תוצאה בתחרות. מספר שלם.

#### הנחות על קובץ הקלט:

- ניתן להניח שמבנה הקובץ תמיד יהיה תקין ושכל המידע בו נכון.
- מתחרה שמפורטת עבורו תוצאה בתחרות כלשהי, תפורט עבורו גם המדינה אותה מייצג. כלומר, אם מתחרה מופיע בשורה המתחילה ב- competition אז הוא יופיע גם בשורה עבור competitor.
- מתחרה מוגדר פעם אחת בלבד בקובץ.
- אין חשיבות לסדר בין השורות. בפרט, שורות המגדירות תוצאה של מתחרה בתחרות יכולות להופיע בקובץ לפני שורות המגדירות את המתחרה עצמו ולהפך.
- פרמטרים מורכבים ממחרוזת יחידה, ובין כל שני פרמטרים מפריד רווח בודד (white space).
- מתחרה יכול להתחרות במספר תחרויות שונות.

- ייתכנו שורות בהן אותו מתחרה מנסה להתחרות באותה תחרות יותר מפעם אחת, על מנת להשיג יותר מדליות עבור מדינתו. מתחרה מסוג זה **נפסל** ואין להתחשב בכל תוצאותיו בתחרות זו.
- במידה ומתחרה **נפסל** בתחרות מסוימת, הוא עדיין יכול להתחרות בתחרות **אחרת**.

### 3.3. הקובץ hw2.py

הקובץ hw2.py הוא קובץ הפייתון שתגישו בתרגיל זה. הקובץ מכיל:

- ה-scope הראשי של התוכנית “\_\_main\_\_”
- פונקציה בשם partA אשר מימשנו עבורכם. הפונקציה משתמשת בפונקציות שתידרשו לממש בחלק א', ומדפיסה נתונים בפורמט אחיד.
- פונקציה בשם partB אשר עליכם להשלים לפי חלק ב' של תרגיל זה.
- שתי פונקציות הדפסה: printCompetitor, printCompetitionResults, ופונקציית עזר: keySortCompetitor. את הפונקציות הנ"ל מימשנו עבורכם והן משמשות בפונקציה partA.
- פונקציות אשר עליכם לממש:
- readParseData - הסבר בסעיף 3.4
- calcCompetitionResults - הסבר בסעיף 3.5

### 3.4. קריאת קובץ הקלט וניתוח הטקסט

ממשו את הפונקציה בעלת החתימה הבאה:

```
readParseData(file_name)
```

הפונקציה מקבלת כקלט שם של קובץ קלט (file\_name) שהינו קובץ טקסט, קוראת מהקובץ ומנתחת אותו. הפונקציה מחזירה רשימה המכילה נתוני השתתפות של מתחרים בתחרויות. כל רשומה (איבר ברשימה) היא מילון המכיל:

```
{
  "competition name": competition_name,
  "competition type": competition_type,
  "competitor id": competitor_id,
  "competitor country": competitor_country,
  "result": result
}
```

הערות נוספות:

- כפי שרשום לעיל, כל מפתח במילון הוא מחרוזת המאפיינת את הפרמטר ששייך אליו.
- פירוש הפרמטרים וטיפוסם הוא לפי סעיף 3.2.
- במידה ומתחרה משתתף יותר מפעם אחת בתחרות מסוימת, אז עבור תחרות זו הוא יופיע ברשימה כמספר השתתפויותיו בתחרות הזו.
- ניתן להניח כי קובץ הקלט נמצא בתיקיה הנוכחית.
- לפענוח קובץ הקלט, מומלץ להשתמש בפונקציה `split` של פייתון, אשר מפרקת מחרוזת למילים. ניתן לקרוא מידע נוסף על הפונקציה `split` [כאן](#).
- הדפסת נתוני המתחרים ומיון שלהם עבור ההדפסה מומשו בפונקציה `partA`, לפי הפלט של הפונקציה `readParseData`. נתוני המתחרים ממוינים לפי סדר לקסיקלי של `competition_name`, וסדר משני לפי ערך מספרי של `result`. סדר המיון חשוב והמימוש שלכם ייבדק לפי סדר שורות תואם בהדפסה.

### 3.5. מימוש הפונקציה לחישוב תוצאות תחרות

ממשו את הפונקציה בעלת החתימה הבאה:

```
calcCompetitionResults(competitors_in_competitions)
```

הפונקציה מקבלת כקלט רשימה, כך שכל רשומה בה היא מילון המכיל את נתוני ההשתתפות של מתחרה בתחרות מסוימת (זהה לפלט של הפונקציה `readParseData`):

```
{'competition name': competition_name, 'competition type': competition_type,
'competitor id': competitor_id, 'competitor country': competitor_country,
'result': result}
```

הפונקציה מחשבת את תוצאות התחרויות באולימפיאדה ומחזירה רשימה של תוצאות התחרויות, כך שכל תא ברשימה מכיל (בסדר הזה):

```
[competition_name, winning_gold_country,
winning_silver_country, winning_bronze_country]
```

פירוש הפרמטרים (כולם מחרוזות):

- `competition_name` – כפי שהגדרנו קודם בתרגיל, שם תחרות.
  - `winning_gold_country` - המדינה הזוכה בזהב ב- `competition_name`.
  - `winning_silver_country` - המדינה הזוכה בכסף ב- `competition_name`.
  - `winning_bronze_country` - המדינה הזוכה בארד ב- `competition_name`.
- שימו לב כי ערך ההחזרה הוא רשימה המכילה רשימות. הסדר של התחרויות ברשימה המוחזרת אינו חשוב, אך יש לשמור על סדר הפרמטרים בכל תא ברשימה כפי שמופיע לעיל.

#### לדוגמא:

בהינתן שעבור התחרות swimming הזוכים הם (מימין לשמאל) USA ,Canada ,Israel  
ועבור התחרות mountain\_bike הזוכים הם (מימין לשמאל) Spain ,USA ,Italy  
הפונקציה calcCompetitionResults תחזיר את הרשימה הבאה:

```
[['swimming', 'Israel', 'Canada', 'USA'],  
 ['mountain_bike', 'Italy', 'USA', 'Spain']]
```

במידה ולא קיים זוכה במדליה (משמע, סיימו באופן חוקי פחות משלושה מתחרים, אך לפחות אחד), יש להציב במקום שם המדינה את המחרוזת 'undef\_country'.

אולם, אם לא סיים אף מתחרה באופן חוקי (משמע, קיימים אפס זוכים במדליה כלשהי בתחרות מסוימת) אז אין צורך לעדכן את סך המדליות באולימפיאדה ולכן אין להחזיר את הנתונים על תחרות זו.

שימו לב לשוני בקביעת המנצח בכל אחד משלושת סוגי התחרויות, כפי שפורט בסעיף 3.1.

#### המלצות למימוש:

- עבור מיון, מומלץ להשתמש בפונקציות sort או sorted שנלמדו בתרגולים. הרחבה על פונקציות אלה ניתן למצוא כאן.

#### הערות נוספות:

- הדפסת נתוני התחרויות ומיון שלהם עבור ההדפסה מומשו בפונקציה partA, ומתבססים על הפלט של הפונקציה calcCompetitionResults. נתוני התחרויות ממוינים לפי סדר לקסיקלי של competition\_name. סדר המיון חשוב והמימוש שלכם ייבדק לפי סדר שורות תואם בהדפסה.

## 4. חלק ב – עבודה עם swig – מימוש תסריט האולימפיאדה

בחלק הזה נתון לכם קוד ספרייה הכתוב בשפת C, שמחשב את המנצח באולימפיאדה על סמך התוצאות של תחרויות בודדות. אתם מתבקשים ליצור ממשק פייתון לספרייה תוך שימוש בכלי SWIG כפי שנלמד בתרגולים, ולאחר מכן להשתמש בממשק שיצרתם כדי לחשב את המדינה המנצחת.

## 4.1. הכנת ממשק python למודול C ע"י SWIG

נתונים לכם שלושת הקבצים הבאים:

- Olympics.h – קובץ ממשק אשר כולל פונקציות שימשו אתכם למימוש התסריט.
- Olympics.o – קובץ בינארי הכולל את מימושו של הקובץ Olympics.h.
- Olympics.i – קובץ המשמש את הכלי SWIG, שהנכם נדרשים להשלימו.

**קובץ הממשק:**

קובץ הממשק נקרא Olympics.h ומכיל 4 פונקציות שימושיות

**Olympics OlympicsCreate();**

פונקציה זו יוצרת אובייקט מסוג אולימפיאדה, ומחזירה אותו.

**void OlympicsUpdateCompetitionResults(Olympics o, const char\*  
gold\_country, const char\* silver\_country, const char\* bronze\_country);**

פונקציה זו מקבלת אובייקט אולימפיאדה ושלוש מדינות (תחילה המדינה של המשתתף שזכה במדליית זהב, לאחר מכן כסף ואז ארד) ומעדכנת את הניקוד הכולל באולימפיאדה של כל מדינה עד כה.

**void OlympicsWinningCountry(Olympics o);**

פונקציה זו מקבלת אובייקט אולימפיאדה ומדפיסה את המדינה המדורגת הכי גבוה בטבלת הניקוד של המדינות (אופן הדירוג נקבע בפונקציה ואין צורך להתייחס אליו בקוד שלכם). יש להשתמש בפונקציה זו אחרי שהפונקציה הקודמת (OlympicsUpdateCompetitionResults) סיימה את כל קריאותיה. כלומר, תשובת פונקציה זו תלויה בתוצאות התחרויות המסופקות תוך שימוש בפונקציה הקודמת.

**void OlympicsDestroy(Olympics o);**

פונקציה זו מקבלת אובייקט אולימפיאדה והורסת אותו, יש לקרוא לפונקציה זו בסוף התסריט שלכם על מנת לשחרר את הזיכרון שהקצאתם בפונקציה הראשונה.

## הקובץ Olympics.i

השלימו את הקובץ Olympics.i והעבירו אותו עם שאר הקבצים המסופקים לשרת. לאחר מכן הריצו את הפקודות הבאות:

1. `swig -python Olympics.i`
2. `gcc -std=c99 -fPIC -c Olympics_wrap.c -I/usr/local/include/python3.6m`
3. `ld -shared Olympics.o Olympics_wrap.o -L/usr/local/include/python3.6m/ -o _Olympics.so`

## 4.2. שילוב Olympics בסקריפט הפייתון

עליכם לממש את המשך הפונקציה **partB** בקובץ **hw2.py**, כך שבהינתן קובץ טקסט, מהסוג שתואר בסעיף 3.2, מדפיסה למסך את המדינה המנצחת. ניתן להניח כי קובץ הקלט נמצא בתיקיה הנוכחית. השתמשו ב `swig`, כפי שנלמד בתרגולים, ועדכנו עבור כל תחרות בה לפחות מנצח אחד את נתוניו באולימפיאדה על ידי הפונקציה `OlympicsUpdateCompetitionResults`. כדי לחשב את המדינה המנצחת באולימפיאדה השתמשו בפונקציה הייעודית `OlympicsWinningCountry` מתוך סקריפט הפייתון.

### הנחיה:

- עליכם לוודא שהטיפוס של כל מחרוזת שאתם מעבירים לפונקציית C הוא `str` (ולא `bytes`), לדוגמה ע"י המרה מפורשת באמצעות הקריאה לפונקציה `str:`

`Olympics.OlympicsUpdateCompetitionResults(o,str(gold), str(silver),...)`

## 5. הגשה

את ההגשה יש לבצע דרך אתר הקורס, תחת `Assignments→HW2→Electronic Submit`, פורמט ההגשה כמו באיור מטה.

- הקובץ `Submission.zip` יכיל את שלושת הקבצים הבאים בלבד:
  - הקובץ `Olympics.i` לאחר השינויים שלכם.



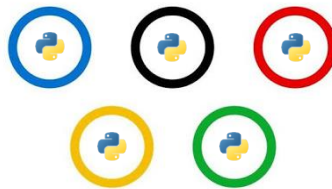
- קובץ hw2.py המעודכן, שמכיל את הפונקציות שהתבקשתם לממש ולהשלים. אין לשנות את הפונקציות וחלקי הפונקציות שמומשו עבורכם!!!

Submission.zip  
 └─ Olympics.i  
 └─ hw2.py

- שימו לב: השתמשו אך ורק בzip (פורמט אחר לא יתקבל). אין חשיבות לשם קובץ zip המוגש.
- שימו לב, אי עמידה במבנה הגשה זה תגרור הורדת נקודות.
- נתון לכם תסריט בדיקה של מבנה ההגשה, הנקרא finalCheck בתיקיית התרגיל בשרת. על מנת לבצע בדיקה זו, העבירו את קובץ zip שלכם לשרת והריצו את הפקודה הבאה:

```
~mtm/public/1920b/ex2/finalCheck hw2.zip
```

- התסריט יבדוק שהיררכיית הקבצים נכונה, ובפרט שמבנה הקובץ שאתם מגישים נכון, ויעביר טסט פשוט וקוד שכתבתם. טסט זה הוא טסט בקרה בלבד ואינו מספק לבחינת התרגיל במלואו!
- הקפידו לבדוק את נכונות הקוד שלכם עם מגוון טסטים. אנו ממליצים גם על שיתוף טסטים בין סטודנטים.
- הטסטים ייבדקו לפי סדר שורות תואם (סדר מיון זהה של השורות בהדפסה).
- אין להוסיף קוד מחוץ לפונקציות או ל\_\_main\_\_.
- מותר להגיש את התרגיל מספר פעמים, רק ההגשה האחרונה נחשבת.
- על מנת לבטח את עצמכם נגד תקלות בהגשה האוטומטית שימרו את קוד האישור עבור ההגשה. עדיף לשלוח גם לשותף. כמו כן שימרו עותק של התרגיל על חשבון ה-cs13 שלכם לפני ההגשה האלקטרונית ואל תשנו אותו לאחריה (שינוי הקובץ יגרור שינוי חתימת העדכון האחרון).
- \*\*\* כל אמצעי אחר לא יחשב הוכחה לקיום הקוד לפני ההגשה.



*Safety Distance*

**בהצלחה!!!**