

הנחיות כלליות

יש לשלוח את הקבצים באמצעות [מערכת ההגשה](#) לפני חלוף התאריך **20/1/20**.

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 6 – ass6

משקל התרגיל מתוך ציון התרגול: 20%.

- בתרגיל זה נעשה שימוש **בהקצאה דינמית**. עליכם לוודא שלא נותר זיכרון שאינו משוחרר (**דליפת זיכרון**) בסיום ההרצה. הנושא הזה נבדק אוטומטית ומדווח לכם בפיידבק המיידי במייל.

בתרגיל זה עליכם לממש את מבנה הנתונים **מילון** (dictionary).

מילון מכיל מיפוי בין מפתחות (keys) לערכים (values). כל מפתח הוא ייחודי במילון, והוא ממופה אל ערך יחיד (אך ייתכנו שני מפתחות שונים הממופים לשני ערכים זהים).

מילון

בקובץ **dictionary.h** מופיעות הצהרות המגדירות את הפעולות הבסיסיות שנדרשות על מנת לממש מילון. עליכם לממש את הפונקציות האלו בקובץ **dictionary.c** (כל פונקציית עזר שתצטוו לממש עבור הפונקציות האלו, תימצא גם כן בקובץ הזה).

אם אתם משתמשים במבני נתונים נוספים, או בקטעי קוד שיש להם הצדקה כיחידה עצמאית, הגדירו אותם **כמודול נפרד**. תכננו את האופן שבו אתם ניגשים לתרגיל מראש, ורק אחר כך התחילו לכתוב את הקוד.

הגדרת הפעולות:

- אתחול (**initDictionary**) – הפונקציה מקצה זיכרון עבור מילון ריק ומחזירה מצביע אליו.
- שחרור (**destroyDictionary**) – הפונקציה משחררת את כל הזיכרון שהוקצה עבור המילון.
- גודל המילון (**sizeofDictionary**) – הפונקציה מחזירה את מספר המפתחות/הערכים במילון.
- הכנסה (**putInDictionary**) – הפונקציה מקבלת מפתח וערך ומכניסה אותם למילון, אם המפתח כבר קיים במילון, הערך המשוך אליו יוחלף בערך החדש.
- חיפוש לפי מפתח (**getFromDictionary**) – הפונקציה מקבלת מפתח ומחזירה את הערך הממופה אליו במילון. אם המפתח לא נמצא, יש להחזיר 0 (אך אין לשמור מיפוי כזה במילון אם לא הוכנס בפירוש).
- הוצאה (**removeFromDictionary**) – הפונקציה מקבלת מפתח ומסירה אותו מהמילון (ואת הערך הממופה אליו). אם המפתח לא קיים במילון, אין לעשות דבר.
- הדפסה (**printDictionary**) – הפונקציה מדפיסה את כל תוכן המילון, ממוין לפי מפתחות.
- יצירה ממערכים (**createDictionaryFromArrays**) – הפונקציה מקבלת שני מערכים בגודל שווה ואת גודלם, ויוצרת מילון על פיהם (דוגמה מפורטת בקובץ ה-main המצורף לתרגיל).

מימוש:

אתם יכולים לבחור את המימוש כרצונכם, אך שימו לב שאין להגדיר מבנים בקובץ H כלל, כדי לשמור על מודולריות.

הנחיה: המימושים הנפוצים הם בעזרת עץ או בעזרת מערך, מומלץ להתבסס על העץ שהוצג בכיתה או על המערך הדינמי מתוך המחסנית שמימשתם. ייתכן שיידרשו התאמות מסויימות (מבחינת שמות הפונקציות, וגם מבחינת המימוש עצמו).

הקובץ main.c

בקובץ הזה מופיעה פונקציית ה-main שמפעילה פונקציות המתפעלות את המודולים השונים.

הקובץ מיועד לבדיקות אישיות שלכם, וייתכן שיעודכן מדי פעם.

שימו לב שאין להגיש קובץ main או לכלול בקבצים שלכם פונקציית main.

makefile

בתרגיל זה עליכם להגיש את כל הקבצים שנדרשים לצורך המימוש שלכם, ובנוסף קובץ [makefile](#).

פקודת הקימפול בתרגיל הזה תהיה **make** (בנוכחות הקובץ הזה).

עליכם לדאוג לקימפול תקין של כל המודולים בתוכנית שלכם, ולהניח את קיומו של קובץ [main.c](#) שיקומפל עם כל המודולים על מנת ליצור קובץ ריצה תקין בשם [a.out](#) – שאותו תפעיל הפקודה **make run**.

פלט עבור קובץ ה-main המצורף

```
Size: 0
Dictionary: {}
Size: 1
Dictionary: {[1:10]}
Size: 1
Dictionary: {[1:2]}
Size: 0
Dictionary: {}
Size: 0
Dictionary: {}
43: 0
Size: 0
Dictionary: {}
Size: 8
Dictionary: {[[-61:2][[-12:7][0:10][7:10][12:10][14:-9][43:-5][97:10]}
43: -5
Size: 8
Dictionary: {[[-61:2][[-12:7][0:10][7:10][12:10][14:-9][43:-5][97:10]}
Size: 6
Dictionary: {[[-61:2][0:10][7:10][12:10][14:-9][97:10]}
```