

תרגיל בית 4 - C++ בסיסי

מועד ההגשה:	יום ה', 12/1/2017 בשעה 23:59
האחראי על התרגיל:	אסף קסל
	sasafca@campus

רשת חברתית

בתרגיל זה, נבנה רשת חברתית בסיסית מאד. משתמשים ברשת יזוהו עפ"י כתובת הדואר שלהם. בנוסף יהיה להם גם שם וסיסמא. נרצה לאפשר שני סוגי משתמשים, עוקבים (followers) ומובילים (leaders). עוקב (follower) יכול ליצור קשרי חברות, לשלוח הודעות, ו"לעקוב" אחרי מוביל. מוביל יכול לעשות את כל הפעולות של עוקב ובנוסף להפיץ הודעות לכל רשימת העוקבים שלו.

תיאור התרגיל מחולק לחלקים באופן הבא:

1. דרישות – תיאור של פעולת פקודות המערכת.
2. דגשים ומגבלות – הסבר איזה קוד כבר סופק, מה ניתן לשנות/להוסיף ודגשים כלליים.
3. הדרכה – הצעה (לא מחייבת) לתכנון מבנה הקוד שישמש לפתרון התרגיל
4. תרגילון Bash
5. הנחיות הגשה

1. דרישות

נתחיל מדרישה כללית ממבני האיחסון (containers) בהם תשתמשו. במהלך התרגיל תצטרכו להשתמש במבני איחסון עבור עוקבים, מובילים והודעות. עליכם להשתמש לכך במבנה מסוג רשימה מקושרת אשר מוסיף איברים חדשים **תמיד** לתחילת הרשימה. **דבר זה חשוב** כדי שסדר ההדפסה שלכם יהיה תואם לזה שאנו השתמשנו בו.

כעת נתאר את רשימת הפקודות שבהן תומכת התוכנה. לכל פקודה יש אוסף תנאים אשר מגדיר האם היא תצליח או תכשל. כדי למנוע בלבול בצורת ההדפסה, סיפקנו לכם את כל שורות ההדפסה הדרושות כדי להודיע על הצלחה או כישלון (יוסבר בדגשים ומגבלות). לכן, בתיאור הפקודות כאן נציין רק את התנאים להצלחה וכישלון מבלי לציין מה יודפס בכל מקרה. להלן רשימת הפקודות:

1. AdminLogin <password> פעולה – התחברות של מנהל הרשת.
הצלחה - כאשר <password> מתאימה לסיסמאת הרשת אשר מסופקת בזמן יצירתה.
כישלון - אחרת.
2. Login <email> <password> פעולה – התחברות של משתמש לרשת (מוביל או עוקב).
הצלחה – כאשר המשתמש המתאים ל <email> קיים <password> מתאימה לסיסמא שלו.
כישלון – אחרת.
3. Logout פעולה – מנתקת את המשתמש הנוכחי מהרשת.
הצלחה – כאשר משתמש או מנהל מחוברים.
כישלון – אחרת.
4. CreateLeader <name> <email> <password> פעולה – יצירת מוביל עם הפרמטרים המסופקים והוספתו לרשת.

הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל

מבוא למערכות תוכנה

הצלחה – כאשר מנהל הרשת מחובר ולא קיים משתמש אחר (עוקב או מוביל) עם כתובת מייל

זהה ל `<email>`.

כישלון – אחרת.

5. `DeleteUser <email>`

פעולה – מוחקת משתמש ברשת (מוביל או עוקב) אשר מתאים לכתובת המייל `<email>`. יש לעדכן את חברי ומובילי המשתמש על כך שהוא לא קיים יותר.

הצלחה – כאשר מנהל הרשת מחובר ומשתמש עם כתובת המייל `<email>` קיים.

כישלון – אחרת.

6. `BroadcastMessage <subject> <content>`

שימו לב! יש להסתכל בקובץ `main.C` כדי לראות את פורמט השימוש בפקודה.

פעולה – שולחת הודעה עם נושא `<subject>` ותוכן `<content>` לכל העוקבים של המוביל אשר מחובר לרשת.

הצלחה – כאשר מוביל מחובר לרשת.

כישלון – אחרת.

7. `CreateFollower <name> <email> <password>`

פעולה – יצירת עוקב עם הפרמטרים המסופקים והוספתו לרשת.

הצלחה – כאשר לא קיים משתמש אחר (עוקב או מוביל) עם כתובת מייל זהה ל `<email>`.

כישלון – אחרת.

8. `ShowFriendRequests`

פעולה – הדפסה של בקשות החברות של המשתמש אשר מחובר לרשת.

הצלחה – כאשר משתמש מחובר לרשת.

כישלון – אחרת.

9. `ShowFriendList`

פעולה – הדפסה של רשימת החברים של המשתמש אשר מחובר לרשת.

הצלחה – כאשר משתמש מחובר לרשת.

כישלון – אחרת.

10. `SendFriendRequest <email>`

פעולה – הוספת בקשת חברות למשתמש המזוהה ע"י `<email>` מהמשתמש אשר מחובר לרשת.

כישלון – כאשר לפחות אחד מהבאים מתקיים:

- אין משתמש מחובר לרשת
- אין משתמש עם כתובת מייל `<email>`
- ניסיון לשלוח בקשת חברות לעצמך
- המשתמשים כבר חברים
- בקשת חברות כבר נשלחה למשתמש `<email>`

הצלחה – אחרת.

11. `AcceptFriendRequest <email>`

פעולה – אישור בקשת החברות של המשתמש `<email>` ע"י המשתמש המחובר. החברות הינה הדדית ולכן יש צורך לעדכן את שני המשתמשים (המבקש והמסכים) על החברות ולהסיר את בקשות החברות אחד של השני.

הצלחה – כאשר יש משתמש מחובר ויש לו בקשה ממשתמש המתאים לכתובת המייל

`<email>`.

הטכניון - מכון טכנולוגי לישראל
הפקולטה להנדסת חשמל
מבוא למערכות תוכנה

כישלון – אחרת.

12. RemoveFriend <email>

פעולה – ביטול החברות בין המשתמש המחובר למשתמש המתאים לכתובת המייל <email>. החברות הינה הדדית ולכן יש צורך לעדכן את שני המשתמשים (המבקש והמסכים) על ביטול החברות.

הצלחה – כאשר יש משתמש מחובר ויש לו משתמש חבר אשר מתאים לכתובת המייל <email>.

כישלון – אחרת.

13. ShowMessageList

פעולה – מציגה תקציר של רשימת ההודעות של המשתמש אשר מחובר לרשת. כל הודעה תודפס ע"י המתודה Display(int messageNum) אשר סופקה לכם (ראו בהמשך).

הצלחה – כאשר יש משתמש מחובר לרשת.

כישלון – אחרת.

14. ReadMessage <messageNumber>

15. פעולה – מדפיסה למסך את ההודעה שמספרה <messageNumber> ומסמנת שהיא נקראה. מספר ההודעה איננו חלק מההודעה אלא מיקומה ברשימת ההודעות של המשתמש והוא מודפס כאשר מבצעים את הפעולה ShowMessageList. ההדפסה מבוצעת ע"י הפעולה Read() אשר סופקה לכם (ראו בהמשך).

הצלחה – כאשר יש משתמש מחובר ומספר ההודעה תקין (גדול מאפס וקטן או שווה למספר ההודעות ברשימת ההודעות של המשתמש)

כישלון – אחרת.

16. SendMessage <email> <subject> <content>

שימו לב! יש להסתכל בקובץ main.C כדי לראות את פורמט השימוש בפקודה.

פעולה – שולחת הודעה עם נושא <subject> ותוכן <content> למשתמש עם כתובת מייל <email>.

הצלחה – כאשר יש משתמש מחובר ויש לו חבר ברשימת החברים שמתאים לכתובת המייל <email>.

כישלון – אחרת.

17. Follow <email>

פעולה – מוסיפה את המשתמש המחובר לרשימת העוקבים של המוביל המתאים לכתובת המייל <email>.

הצלחה – כאשר יש משתמש מחובר, קיים מוביל עם כתובת מייל <email> והמשתמש המחובר איננו כבר עוקב של מוביל זה.

כישלון – אחרת.

18. FindUser <partialName>

שימו לב! בקובץ SocialNetwork.C סופק לכם קוד חלקי המשמש לפקודה זאת.

פעולה – מדפיסה את רשימת העוקבים אשר שמם (name_) מכיל את המחרוזת <partialName> ולאחר מכן חוזרת על הפעולה עבור רשימת המובילים.

הצלחה – תמיד.

כישלון – אף פעם.

19. Exit

פעולה – סגירת (מחיקת) הרשת וסיום ריצת התוכנה (כולל שחרור זיכרון)

הצלחה – תמיד.

כישלון – אף פעם.

2. דגשים ומגבלות

נסקור את הקבצים הנתונים לכם ומה בהם מותר לשנות:

1. defs.H – קובץ הגדרות שבעיקר מכיל את פורמט ההדפסה עבור הצלחה וכישלון של פקודות התוכנה. בחלק מהפקודות יש שימוש ישיר בשמות משתנים. ניתן לשנות את השמות כרצונכם אך בטוח יותר להתאים את השמות בקוד שלכם. השימוש בפורמטי ההדפסה נעשה ע"י:
`cout << <FORMAT> << endl;`
ישנם שני מקרים חריגים בהם לא שמים cout אך זה מצויין עבורם בקובץ זה.
דוגמא לשימוש: כאשר RemoveFriend מצליחה יש לכתוב,
`cout << REMOVE_FRIEND_SUCCESS << endl;`
ובכישלון נכתוב,
`cout << REMOVE_FRIEND_FAIL << endl;`
2. Follower.C – מיועד למימוש העוקבים (נתון ריק). השתמשו בו לפי ראות עיניכם.
3. Follower.H – מיועד להצהרת מחלקת העוקבים. נתונה הצהרה חלקית אך אתם יכולים לשנות אותה.
4. Leader.C/H – מיועד להצהרת/מימוש מחלקת המובילים (נתון ריק). השתמשו בו לפי ראות עיניכם.
5. Lists.C/H – קובץ שיועד להצהרת/מימוש מחלקות של רשימות מקושרות שמכילות עוקבים או מובילים או הודעות.
6. main.C – קובץ התוכנית הראשי. מומלץ לקרוא ולהבין אותו אך אין לשנות אותו.
7. Message.C/H – מיועד להצהרת ומימוש ההודעות ותיבת ההודעות. מחלקת ההודעה (Message) נתונה לכם במלואה ואין צורך לגעת בה. בנוסף נתונה הצעה לממשק של מחלקת MessageBox אשר יכולה לשמש כתיבת הודעות. מותר לכם לשנות את הקבצים לפי ראות עיניכם.
8. SocialNetwork.C – קובץ המימוש של מחלקת SocialNetwork. נתון רק מימוש חלקי של פעולת FindUser ואת השאר יש להשלים. המימוש ניתן גם כדי שפלט התוכנית יהיה מדויק וגם כדי לתת את אופן הבדיקה של הכלה בין מחרוזות.
9. SocialNetwork.H – קובץ ההצהרה של מחלקת SocialNetwork. נתונות כל פונקציות הממשק של המחלקה לא ניתן לשנות פונקציות אלו. אבל, ניתן להוסיף פונקציות ומשתנים שהם private.

דגשים:

1. מנהל הרשת איננו משתמש ולכן אין לו אפשרות לבצע פעולות של משתמש אלא רק הוספה של מובילים ומחיקה של משתמשים.
2. בכל רגע נתון רק משתמש אחד מחובר לרשת. כלומר, התחברות של משתמש אחד מנתקת את האחר. בנוסף, גם התחברות של מנהל הרשת תגרום לניתוק המשתמש הקודם וחיבור משתמש ינתק את מנהל הרשת.
3. הדפסה של רשימה תתבצע מהאיבר הראשון ברשימה לאחרון.
4. מבנה קוד תקין הוא כזה שאין בו משתנים שהם public.
5. כאשר עובדים בסביבת windows יש לשנות את סיומות קבצי ה C ל cpp. זיכרו כמובן לשנות אותן חזרה ל C לפני שאתם מגישים.
6. מומלץ להשתמש במחלקת string ולא ב char* כפי שעשינו ב C. זה יחסוך לכם המון זמן.
7. חובה להשתמש בדגלי הקימפול הבאים (ב t2):
`-g -Wall -std=c++0x`

3. הדרכה

חלק זה יכיל הצעה כללית לתכנון מבנה הקוד של התרגיל. הוא לא מחייב אך מומלץ לקרוא אותו. הסיבה לכך היא שמטרת התרגילים היא לא רק שהתוכנית תעבוד אלא שתהיה כתובה לפי העקרונות הנלמדים בקורס אשר מהווים את הבסיס לפיתוח תוכנה. הסיבה שהדרכה זאת לא מחייבת היא כי ישנן דרכים רבות לפתרון וחשוב ללמוד כיצד לתכנן את מבנה הקוד שלכם. בכל אופן, תיאור המבנים יינתן בנקודות כלליות ולכן יהיה עליכם לחשוב על הפרטים המדויקים של המימוש.

הטכניון - מכון טכנולוגי לישראל
הפקולטה להנדסת חשמל
מבוא למערכות תוכנה

חלק 1 – Lists

כפי שכבר צויין בתחילת התרגיל, נצטרך להשתמש ברשימות מקושרות כדי לאחסן עוקבים, מובילים או הודעות. ניתן לעשות זאת באחת מכמה דרכים אך הבסיס לכל הדרכים זהה. אפשרויות לדוגמא הן:

1. שימוש ב template. האופציה המומלצת אך עוד לא למדתם אותה.
 2. בניית רשימה שאיבר המידע שלה מכיל גם עוקב גם מנהיג וגם הודעה.
 3. בניית שלוש רשימות נפרדות (אחת לכל סוג אחסון). כלומר, לשכפל קוד.
- להלן הצעה לפעולות של מחלקת רשימה שיכולה להכיל איברים מהצורה של אחת האפשרויות שהוצגו. מידע: מצביע לתחילת הרשימה, מצביע למיקום נוכחי ברשימה

פעולות:

1. הזזת המיקום הנוכחי לתחילת הרשימה.
 2. הוספת איבר לתחילת הרשימה.
 3. מחיקת האיבר במיקום הנוכחי.
 4. הזזת המיקום הנוכחי לאיבר הבא.
 5. החזרת הערך במיקום הנוכחי.
 6. החזרת מספר האיברים ברשימה.
- פעולות אלו מספיקות כדי לבצע את משימות התרגיל (ובאופן כללי כל משימה).

חלק 2 – MessageBox

זהו המשך ישיר של הרשימות המקושרות.

מידע: רשימה מקושרת של הודעות.

פעולות: (נתונות בקבצי הקוד)

1. הוספת הודעה.
2. קריאת הודעה.
3. הדפסת תקציר כל ההודעות.
4. החזרת מספר ההודעות שלא נקראו.
5. החזרת מספר ההודעות הכולל.

חלק 3 – Follower

מחלקת העוקבים היא מהעיקריות שבתרגיל. מומלץ לבדוק את פעולתה היטב לפני שממשיכים בתרגיל. מידע: פרטי העוקב (שם, מייל, סיסמא), רשימת חברים, רשימת בקשות חברות, רשימת הודעות.

פעולות:

1. החזר שם.
2. החזר כתובת מייל.
3. בדוק סיסמא.
4. הצג בקשות חברות.
5. הצג רשימת חברים.
6. הוסף בקשת חברות.
7. אשר בקשת חברות (והסר בקשות חברות מתאימות).
8. הסר חבר.
9. החזר מספר בקשות חברות.
10. הצג רשימת הודעות.
11. הוסף הודעה.
12. קרא הודעה.
13. שלח הודעה.
14. החזר מספר הודעות שלא נקראו.

חלק 4 – Leader

המשך ישיר של Follower.

מידע: יורש מ Follower, רשימת עוקבים.

הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל

מבוא למערכות תוכנה

פעולות:

1. הוסף עוקב.
2. הסר עוקב.
3. החזר מספר עוקבים.
4. שלח הודעה לכל העוקבים.

חלק 5 – SocialNetwork

מבנה הרשת החברתית. פעולותיו כבר סופקו לכם בקוד ולכן נציין רק מידע. מידע: פרטי הרשת (שם רשת, סיסמאת מנהל המערכת), רשימת עוקבים, רשימת מובילים, מצביעים ודגלים לציון מצב הרשת (מי המשתמש המחובר? האם יש משתמש מחובר? האם הוא מוביל? האם מנהל המערכת מחובר?).

4. תרגילון Bash

כתבו סקריפט בשם `change_extensions`, בן 8 שורות לכל היותר (לא כולל `#!/bin/bash` השורה) ולא כולל שורות תיעוד) המקבל שלושה פרמטרים. שם תיקייה, ושני סיומות קבצים. הסקריפט משנה את הסיומות של כל הקבצים בתיקייה שהתקבלה הנגמרים בסיומת שהתקבלה כפרמטר השני, לסיומת שהתקבלה כפרמטר השלישי. לדוגמה, השורה

```
change_extensions dir cpp C
```

תשנה את הסיומות של כל הקבצים בתיקייה `dir` שנגמרים ב-`cpp` "לסיומת `C`". אין צורך לבצע בדיקת פרמטרים.

הערה: קראו באינטרנט על מציאת תבניות במשתנים באמצעות האופרטורים

```
${var#Pattern}
${var##Pattern}
${var%Pattern}
${var%%Pattern}
```

למשל, ניתן לקרוא על כך כאן:

<http://tldp.org/LDP/abs/html/parameter-substitution.html>

הערה 2: הסקריפט הנ"ל שימושי כאשר מעבירים קבצים שנכתבו ב-`windows` ל-`unix`.

5. הנחיות הגשה

1. קבצי קוד חלקיים, וכן קבצי קלט ופלט לדוגמה, נמצאים בתיקייה:
`~eesoft/hmw/hmw4`
לפני תחילת העבודה, הורידו את הקבצים לחשבונכם באמצעות הפקודה:
`cp ~eesoft/hmw/hmw4/* .`
2. עברו היטב על הוראות ההגשה של תרגילי הבית המופיעים באתר טרם ההגשה! ודאו כי התכנית שלכם עומדת בדרישות הבאות:
 - התכנית קריאה וברורה
 - התכנית מתועדת היטב לפי דרישות התיעוד המופיעות באתר
 - התכנית מתקמפלת עם הדגלים `-std=c++0x` -`Wall` -`g` ללא שגיאות וללא `warnings` כלל
 - התכנית רצה ללא **דליפות זיכרון** וגישות לא חוקיות לזיכרון כלל (בדיקה באמצעות `valgrind`)
 - התכנית נותנת פלט **זהה לחלוטין** לפלט הצפוי על כל קבצי הקלט שסופקו (בדיקה באמצעות פקודת `diff` על קבצי הפלט)

הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל

מבוא למערכות תוכנה

• קובץ ה-makefile יוצר קובץ הרצה בשם הנדרש

3. יש להגיש קובץ tar יחיד המכיל את **כל הקבצים** שאתם נדרשים להגיש **ואותם בלבד** – ללא תתי-תיקיות. ודאו כי לא שכחתם את קובץ readme המכיל את פרטי הסטודנטים, וכן את ה-makefile במידה ונדרשתם.

4. שאלות בנוגע לתרגיל יש להפנות לפורום התרגיל ב-moodle בלבד – ניתן לשלוח שאלות במייל **למתרגל האחראי על התרגיל בלבד**, ורק במידה והשאלה מכילה פתרון חלקי.

5. סיכום מפרט התרגיל:

סעיף	תיאור
נושא התרגיל	C++ בסיסי
תאריך ההגשה	יום ה', 12/1/2017 בשעה 23:59
המתרגל האחראי על התרגיל	אסף קסל
תיקייה המכילה קבצים לשימוש הסטודנטים	sasafca@campus ~eesoft/hmw/hmw4
קבצי הקוד הנתונים	defs.H Follower.H Follower.C Leader.H Leader.C Lists.H Lists.C main.C Message.H Message.C SocialNetwork.H SocialNetwork.C
קבצי הקלט והפלט הנתונים	1.MamatNet.in 1.MamatNet.out
הקבצים שיש להגיש	readme makefile defs.H Follower.H Follower.C Leader.H Leader.C Lists.H Lists.C main.C Message.H Message.C SocialNetwork.H SocialNetwork.C change_extensions
שם תכנית ההרצה הדרושה (הנוצרת ע"י makefile)	MamatNet
דגשים מיוחדים	

בהצלחה!