האוניברסיטה העברית בירושלים

בית הספר להנדסה ולמדעי המחשב ע"ש רחל וסלים בנין

Programming Workshop in C & C++ - 67312

תרגיל בית 3 – שפת C
סמסטר חורף, 2020/21

מועד פרסום התרגיל: תאריך - 19/11/2020, יום חמישי
מועד הגשת התרגיל: תאריך - 03/12/2020, יום חמישי, בשעה - 23:25

1 רקע

מגפת הקורונה הגיעה לישראל בפברואר 2020 והחלה להתפשט בקצב מסחרר. מרכזי בילוי, קניונים, מסעדות וחללים סגורים התגלו כמוקדי ההדבקה המרכזיים ונאלצו לסגור באופן מיידי. בחודש יולי באותה שנה, החלו מוסדות אלו לפתוח בחזרה באופן הדרגתי את עסקיהם אך ללא פתרון ממשי כיצד להתמודד עם חולים פוטנציאליים באותם חללים סגורים כך שמדינת ישראל לסגר מהר מאוד כבר במהלך חודש ספטמבר לכל תקופת החגים.

מדינת ישראל החליטה לפנות למוסד האקדמי המוביל בישראל - האוניברסיטה העברית על מנת שתפתח תוכנה שתנתח מידע על מפגשים בין אנשים בקצב, ותקבע במהירות אילו אנשים עם חשש להידבקות בקורונה ובכך לקטוע את שרשראות ההדבקה ההמוניות. האוניברסיטה העברית החליטה להטיל את המשימה על הסטודנטים האיכותיים ביותר באוניברסיטה על מנת שאלו יסייעו בפיתוח התוכנה היעילה, המהירה והאלגנטית ביותר בשפת התכנות הטובה ביותר - שפת C.

2 התוכנה Spreader Detector (מוטיבציה)

התוכנה Spreader Detector הינה תוכנה מתוחכמת ורב מערכתית שנועדה לזהות שרשראות הדבקה פוטנציאליות, להעריך את הסיכוי שאדם ידבק בקורונה ולעדכן את הנדבקים הפוטנציאליים במהירות שיא וביעילות. בתרגיל זה אתם תממשו חלק ממערכת זו.

התוכנה פועלת באופן הבא:
(1) היא מקבלת סרטונים של אותם אנשים שזוהו כבעלי פוטנציאל להיות נשאי קורונה.
(2) התוכנה מחלצת את המידע מתוך הסרטונים באופן הבא: את מי פגשו הנשאים, את מי פגשו אלו שהם פגשו וכך הלאה עד שמזהים את כל האנשים שהופיעו בסרטונים ויודעים מי פגשו את מי ומתי.
(3) התוכנה מעבירה את המידע לבסיס נתונים מתאים.
(4) התוכנה מעריכה את הסיכוי של כל אחד מהמועמדים להידבק מהנשא המזוהה בעזרת נוסחה מתמטית (בהמשך).
(5) מעדכנת את הנדבקים הפוטנציאליים בהתאם למידע הזה ולרמת החומרה.

התוכנית שאתם תכתבו בתרגיל זה, תממש את החלקים 3-5 בתוכנה. כלומר, אתם תקבלו מידע שחולץ מסרטוני הוידאו של אדם שזוהה כנשאי קורונה (כתוב על קבצי טקסט) , תנתחו אותו, תעבדו אותו ותקבעו כיצד לטפל בנדבקים הפוטנציאליים.

3 התוכנית SpreaderDetector (מימוש)

על מנת לממש את התוכנה בקלות ובצורה "נקיה" יותר, בחרנו לחלק עבורכם את התוכנה ל3 ממשקים שונים אשר לכל אחד API נפרד.
שימו לב – בעת בדיקת התרגיל, נבדוק כל ממשק באופן נפרד ובלתי תלוי מהאחרים (unit test) וגם נבדוק את כל הממשקים באופן אינטגרטיבי (integration test).
שימו לב – נתונים לכם headers (קבצי h) עבור כל אחד מהממשקים. בקבצים הללו נתונים חתימות הפונקציות וחתימות הstructs.
אסור לכם לשנות את חתימות הפונקציות הנתונות שם בשום פנים ואופן, **אינכם מגישים קבצים אלו**. המימושים צריכים להופיע בקבצי הc.
אי מעקב אחר הAPI בקבצים האלו יגרור הורדת נקודות ללא אפשרות לערעור.
הממשקים (APIs) אותם תממשו:
1) Person – אובייקט המתאר אדם יחיד המתואר בקובץ האנשים.
2) Meeting – אובייקט המתאר מפגש יחיד בין שני אנשים, כמו שמתואר בקובץ המפגשים.

3) SpreaderDetector – מעטפת לשני האובייקטים הקודמים יחד, כולל את כל האנשים המופיעים בקובץ האנשים ואת כל המפגשים מקובץ המפגשים.

התוכנה היא חלק מפרויקט לאומי רחב ונחוץ, כלומר, התוכנה הולכת לקבל קלט גדול מאוד והיא צריכה לפעול במהירות וביעילות **מיטבית**. נתחו את זמני הריצה של הפונקציות שלכם ונסו לשפר אותם ככל הניתן.

4 קלט-פלט

בתרגיל זה נעבוד עם קבצי קלט ופלט.
הנחות אלו תקפות לשני סוגי קבצי הקלט וקובץ הפלט:
* ניתן להניח ש**אורך שורה מקסימלי** בקבצי הקלט הינו 256 תווים (זה מוגדר עבורכם בקבוע).
* **חל איסור לקרוא את הקובץ פעמיים** - קריאת הקובץ פעמיים תוביל להורדת נקודות. (הוראה זו כוללת כמובן שימוש בפונקציות כדוגמת fseek). למען הסר ספק, הכוונה היא שהסמן (cursor) יכול לנוע רק בכיוון אחד, ולא ניתן להשיב אותו לאחור.
* הפרמטרים זמן ומרחק תחומים בגבולות שיוגדרו בקובץ **Constants.h**, שימו לב - אין לכם להסתמך על ערכי הקבועים הנתונים בשום שלב של הפתרון שכן אנו יכולים לשנות אותם מכיוון שאת הקובץ הנ"ל אינכם מגישים. כלומר, השתמשו בקבועים ולא בערכיהם!.
* החזקה של יותר מדי זיכרון, שאינו בשימוש, נחשבת לשגיאה שלא ניתן לערער עליה.

4.1 קלט

1. **קובץ People** - קובץ זה יכלול את הפרטים המזהים של האנשים אשר נקלטו במהלך הסרטונים.
מבנה הקובץ יהיה כך שכל שורה תייצג פרטים מזהים של אדם יחיד. מבנה השורה יהיה:
<Person Name> <Person ID> <Person age> <SICK ?>\n

הסבר:
* **שם האדם** - מחרוזת רציפה של התווים הבאים A-Z, a-z ומקף ("-"). אין צורך לבדוק את תקינות המחרוזת (במילים אחרות כדי לוודא שאתם לא נלחצים, ניתן להניח שהמחרוזת תקינה וחוקית), לא ניתן להניח דבר לגבי אורך המחרוזת.
* **מספר זהות** - רצף באורך 9 ספרות כלשהן. כמו בשם האדם, אין צורך לבדוק את תקינות המספרים (במילים אחרות כדי לוודא שאתם לא נלחצים, ניתן להניח שמספר הזהות תקין וחוקי), המספר לא יתחיל ב0.
* **גיל** - מספר, לא בהכרח שלם, כלשהו בין 0 ל-120 (כולל), כמו בקודמים, ניתן להניח כי הוא תקין.
* **אינדיקטור מחלה** – במקרה והאדם בשורה כלשהי הוא אכן האדם החולה, תופיע גם המחרוזת SICK בסוף השורה (אחרת, לא יופיע כלום). ניתן להניח כי המחרוזת תקינה.

הנחות ודגשים:
* לא ניתן להניח שתעודות הזהות יהיו יחידות. במידה והופיעו שני אנשים עם ת"ז זהות, עליכם להכניס את הראשון ולהתעלם מהשני.
* הנכם יכולים להניח שיהיה רק אדם אחד מזוהה כחולה בקובץ נתון.
* ניתן להניח כי מבנה השורה תקין. כלומר, הנתונים יוצגו בסדר הנכון, מופרדים ברווחים.

2. **קובץ Meetings** - קובץ זה יכלול את הפרטים של מפגשים שתועדו בסרטונים. מבנה כל שורה יהיה:
<person1_id> <person2_id> <distance> <measure>\n

הסבר:
* **שני המספרים הראשונים** - תעודות הזהות של האנשים שנפגשו.
* **המספרים distance, measure -** מתארים את המרחק בו שני האנשים היו אחד מהשני ומה היה משך המפגש, שני מספרים אלו יהיו חיוביים בהחלט אך יכולים להיות שברים. המספרים יהיו חיוביים ממש.

דגשים:
* ניתן להניח כי לא יכול להיווצר "מעגל" - כלומר מצב כזה: A חשוד שהדביק את B, B חשוד שהדביק את C ו-C חשוד שהדביק את A.
באופן כללי, **ניתן להסתכל על הקלט כעץ פורש ללא מעגלים.** כלומר: המדביק יהיה השורש, האנשים שהוא נפגש איתם יהיו בניו, וכך הלאה. ובנוסף, אין קודקוד שמצביעה עליו יותר מקשת אחת.
* **אדם נתון יכול לפגוש יותר מאדם אחד** (וגם יותר מ2).
*הנכם יכולים להניח כי מספרי תעודות הזהות תקינים, אך עליכם לוודא שאלו מופיעים בקובץ האנשים, במידה ולא – הפגישה נוצרה כתוצאה מתקלה בזיהוי, התעלמו ממנה.

4.2 פלט

בעת קריאה לפונקציה המתאימה בAPI, על התוכנית שלכם להפיק קובץ פלט לפי הפורמט המופיע במודל (מופיע קובץ בנספח ב' של מסמך זה).

5 הפונקציה Crna, מציאת סיכויי ההידבקות ואפיון הטיפול

5.1 הפונקציה Crna

הפונקציה Crna היא פיתוח ייחודי של אחד מחברי הסגל באוניברסיטת **Emzeti-Shem** בארצות הברית. הפונקציה הזו מקבלת מידע על מפגש בין שני בני אדם ומחשבת את הסיכוי שאחד מהם ידביק את השני בקורונה, במידה ואחד מהם נושא את הנגיף. הפונקציה פועלת כך: בהינתן שני אנשים p1,p2 שאנו יודעים שהיו במרחק dist (מטרים) אחד מהשני למשך measure (דקות), ה"score" שידביקו אחד את השני הוא:

$$crna_{p_1,p_2}(distance, measure) = \frac{measure * MinDistance}{distance * MaxMeasure}$$

שימו לב - הפונקציה תחזיר תמיד מספר בין 0 ל-1 (הפרמטר dist תמיד שונה מ0 שכן לא יכולים להיות שני אנשים במרחק 0 מטר אחד מהשני) וזה יהיה ה"score" של אדם להידבק לפי הנוסחה.
שימו לב 2 - המשתנים MIN_DISTANCE ו- MAX_MEASURE הינם שני קבועים הנמצאים בקובץ Constants.h המצורף למשימה.
מבחינת אינטואיציה, MIN_DISTANCE משקף את המרחק הקטן ביותר אותו ניתן לתפוס בסרטון, המשתנה MAX_MEASURE הוא משך הזמן הארוך ביותר בו שני אנשים יכולים להיות אחד בקרבת השני.
נבחין כי:

$$\lim_{\substack{d \to MinDistance \\ t \to MaxMeasure}} crna_{p_1,p_2}(d,t) \to 1$$

כלומר, ככל ששני אנשים מתקרבים למשך יותר זמן - סיכויי ההדבקה שלהם עולים.
ובכיוון השני:

$$\lim_{\substack{d \to \infty \\ t \to 0}} crna_{p_1,p_2}(d,t) \to 0$$

כלומר, יותר מרחק ופחות זמן אחד עם השני מוריד את סיכויי ההדבקה.
ריחוק חברתי!!! (במיוחד בזמן כתיבת התרגיל, כדי שהקוד שאתם כותבים לא ידבק בדמיון מהקוד של החבר/ה ליד).

5.2 חישוב הסיכוי להידבקות

אופן החישוב של סיכויי ההידבקות של אדם צריך להיות באופן הבא:
1. <u>אם אדם נתון היה במפגש עם הנדבק (ה- spreader)</u> - חישוב הסיכוי להידבקות נעשה באופן ישיר ע"י הפונקציה Crna כאשר המדביק הוא P1 והנדבק הפוטנציאלי הוא P2, נציב את הזמן והמרחק בו הם פגשו אחד את השני וזה הסיכוי של P2 להידבק.
2. <u>אם אנחנו מדברים על אדם במעגל שני</u> - נמשיך עם הדוגמה מהסעיף הקודם, P1 הוא הנשא המאומת שלנו, P2 היה במגע ישיר עם P1, וישנו אדם חדש שנתפס בעדשת המצלמה שמו הוא P3, הוא וP2 היו במגע לאחר שP2 היה במגע עם P1 ולכן ייתכן וגם הוא נדבק.
מכיוון שאין נתונים ישירים על פגישה שלו עם P1, הסיכוי שלו להידבקות יחושב באופן הבא:

$$Crna_{p1,p2}(d_1,t_1) * Crna_{p2,p3}(d_2,t_2)$$

שימו לב - מכיוון ששני המספרים קטנים מ-1 מכפלתם קטנה מכל אחד מהמספרים לחוד כלומר הסיכוי להדבקה יורד ככל שמתרחקים בשרשרת.
3. <u>המקרה הכללי</u> - אם P_1 הוא חולה מאומת, P_2 היה במגע עם P_1, וידוע שP_3 היה במגע עם P_2 , ... , וידוע שP_n היה במגע עם P_{n-1} אז הסיכוי של P_n להידבק יהיה :

$$Crna_{p1,p2}(d_1,t_1) * Crna_{p2,p3}(d_2,t_2) * \dots * Crna_{p_{n-1},p_n}(d_n,t_n)$$

*** אופן החישוב אכן נראה מעט מורכב אך שימו לב כי שימוש באלגוריתם שמימשתם בתחילת התרגיל יכול לסייע כאן (זה רמז עבה מאוד).
** תיקון – בפרמטרים האחרונים זה d_{n-1}, t_{n-1}.
** **שימו לב – אדם שגילו מעל הקבוע AGE_THRESHOLD מקבל תוספת 0.08 לחישוב הscore.**

5.3 קביעת הטיפול

אופן החישוב של סיכויי ההידבקות של אדם צריך להיות באופן הבא:
1. אם אדם נתון עם infection rate (חושב ע"י הסעיף הקודם) גבוה מ- MEDICAL_SUPREVISION_THRESHOLD עליכם לסווג אותו כדרוש להשגחה רפואית ולהדפיס את ההודעה המצוינת בקובץ הקבועים עם הפרמטרים המתאימים.
2. אם אדם נתון עם infection rate גבוה מ-REGULAR_QUARNTINE_THRESHOLD עליכם לסווג אותו כדרוש לבידוד ביתי.
3. שאר המקרים לא דורשים בידוד וניתן להשתמש בהודעת CLEAN_MSG
* כל אדם מקבל סיווג יחיד.

6 הרצת התוכנית

את התוכנית שלכם נקמפל יחד עם קובץ main.c חיצוני שאינכם ראיתם. הקובץ ישתמש בפונקציות שונות מתוך הAPI אותו כתבתם. קובץ לדוגמא מופיע בmoodle ובנספח ב' של מסמך זה.

7 הגשת התרגיל

עליכם להגיש קובץ tar בשם ex3.tar והוא יכלול את הקבצים הבאים והם בלבד:

Person.c
Meeting.c
SpreaderDetector.c

לא להגיש את קבצי הh (לא נהיה סלחניים לגבי זה).
שימו לב שהנכם מייבאים אותו ומשתמשים בקבועים הנמצאים בו, אך אינכם מגישים אותו. על הקבועים להיות דינמיים ולהשתנות לפי החלטות הממשלה, ללא השפעה על אופן פעילות התוכנה.

8 הערות ודגשים

- אנא וודאו כי התרגיל שלכם עובר את סקריפט ה-pre-submission ללא שגיאות או אזהרות. הסקריפט זמין בנתיב:

```
~labcc/presubmit/ex3/run <path_to_submission>
```

- עליכם לבדוק כי התוכנית מתקמפלת ללא הערות על מחשבי בית ספר לפי הפקודה הבאה:

```
gcc -Wall -Wextra -Wvla -Werror -g -lm -std=c99 <code_files> <main_file> -o SpreaderDetector
```

- בחלק מהבדיקה האוטומטית תבדקו על סגנון כתיבת קוד. אנא ודאו כי הנכם מבינים את דרישות הcoding style של הקורס במלואן.

- הקפידו על בדיקות של דליפות זיכרון ושאר השגיאות אותן בודק ה-**valgrind**. המשקל אשר יינתן לבדיקות אלו יהיה רב ולכן וודאו זאת היטב.
למען הסר ספק, הvalgrind אמור לרוץ באופן תקין מלא-מלא, כלומר שום שגיאה או הערה מכל סוג שהוא. אם מופיעה הערה כלשהי, גם אם אינה קשורה לדליפת זיכרון באופן ישיר, היא נחשבת כשגיאה ותהיה על כך הורדת נקודות.

- תיעוד – הקפידו על תיעוד הולם של הקוד אותו אתם כותבים.

- נראות הקוד – הקפידו על כל הדגשים התכנותיים – אורך פונקציות, שמות משתנים ושמות פונקציות אינפורמטיביים וכו'.
תהיה בדיקה ידנית ואנחנו נתייחס לדגשים אלה בקפדנות.

בהצלחה!

"I have yet to see a language that comes even close to C"

(Linus Torvalds)

נספח א' – קבצי קלט

1. קובץ People:

```
Ryan 533885365 90
Marcus 992150860 80
Kyle 919462684 70
Vernon 353304652 60
Paula 620434839 100 SICK
Dorothy 673174908 50
```

2. קובץ Meeting:

```
620434839 533885365 10 10
620434839 673174908 10 10
533885365 919462684 10 10
533885365 992150860 10 10
992150860 353304652 10 10
```

3. קובץ קבועים:

```
/**
 * Age threshold.
 * Each age above receives INFECTION_RATE_ADDITION_DUE_TO_AGE extra points to the
infection rate.
 */
#define AGE_THRESHOLD 55
#define INFECTION_RATE_ADDITION_DUE_TO_AGE 0.08

/**
 * Minimal distance two people can be in.
 */
#define MIN_DISTANCE 1.0

/**
 * The length of the video, also the maximal time
 * two people can be seen together.
 */
#define MAX_MEASURE 45.0
```

4. קובץ פלט:

```
Hospitalization: Paula 620434839 100 1.000000.
Quarantine: Ryan 533885365 90 0.102222.
No-Treatment: Marcus 992150860 80 0.082272.
No-Treatment: Kyle 919462684 70 0.082272.
No-Treatment: Vernon 353304652 60 0.081828.
No-Treatment: Dorothy 673174908 50 0.022222.
```

נספח ב' – קובץ main לדוגמא

קובץ לדוגמא ללא שימוש בקבצי קלט:

```
#include "SpreaderDetector.h"

int main() {
  SpreaderDetector *spr = SpreaderDetectorAlloc();

  Person *p1 = PersonAlloc(306356321, "Ofira", 53, 0);
  Person *p2 = PersonAlloc(201336708, "Berko", 50, 0);
  SpreaderDetectorAddPerson(spr, p1);
  SpreaderDetectorAddPerson(spr, p2);

  Meeting *meeting = MeetingAlloc(p1, p2, 10.1, 10);
  SpreaderDetectorAddMeeting(spr, meeting);

  size_t meetings_num = SpreaderDetectorGetNumOfMeetings(spr); // returns 1

  PersonFree(&p1);
  PersonFree(&p2);

  MeetingFree(&meeting);

  SpreaderDetectorFree(&spr);

  return 0;
}
```

קובץ לדוגמא עם שימוש בקבצי קלט (כאן נחשב סיכוי הידבקות לכל אדם בהתאם לפגישותיו ונדפיס לקובץ פלט):

```
#include "SpreaderDetector.h"

int main() {
  SpreaderDetector *spr = SpreaderDetectorAlloc();

  SpreaderDetectorReadPeopleFile(spr, "people_file.in");
  SpreaderDetectorReadMeetingsFile(spr, "meetings_file.in");

  SpreaderDetectorGetInfectionChances(spr);
  SpreaderDetectorPrintRecommendTreatmentToAll(spr,"output_file.out");

  // free meeting
  for (size_t k_i = 0; k_i < spr->meeting_size; ++k_i) {
    MeetingFree(&spr->meetings[k_i]);
  }
  // free people
  for (size_t k_j = 0; k_j < spr->people_size; ++k_j) {
    PersonFree(&spr->people[k_j]);
  }
  SpreaderDetectorFree(&spr);

  return 0;
}
```