



האוניברסיטה העברית בירושלים

בית הספר להנדסה ולמדעי המחשב ע"ש רחל וסלים בנין

Programming Workshop in C & C++ - 67315

מבחן מסכם בשפת C

סמסטר אביב, 2020

מועד הגשת המבחן: תאריך - 04/08/2020, יום שלישי, בשעה - 20:30

משך המבחן - 72 שעות.

1 הנחיות למבחן הבית (מופיע בקובץ ההנחיות)

נבקש לחדד את ההנחיות והנהלים לפתרון מבחן הבית, שמתווספים לנהלי הקורס:

- **אמינות ויושרה אקדמית:** זוהי אינה מטלת קורס. זהו מבחן לכל דבר ועניין. לכן, בהתחשב בעובדה שלא קיימת השגחה על מבחן זה כפי שהייתה לו היינו מקיימים את המבחן בבית הספר, נחמיר בצורה יוצאת מן הרגיל (ביחס לתרגילים בקורס) בבדיקת העתקות. עבדו לבד. **חל איסור לשוחח על המבחן ולחלוק חומר כלשהו הנוגע אליו. בין היתר חל איסור לחלוק אלגוריתמים ורעיונות למימוש; לשתף בהתלבטויות ותהיות; לצפות במסך של המחשב של סטודנט/ית אחר/ת שמחבר/ת את המבחן; בוודאי ובוודאי חל איסור חמור לשתף קוד מכל סוג שהוא, לרבות Unit Testing וכן חל איסור להעתיק קוד מהאינטרנט או מכל מקור חיצוני אחר** בדיוק כפי שנעשה במבחן המתקיים בבית הספר (מדובר ברשימה שאינה ממצה – הבנתם את העיקרון). כל שאלה הנוגעת לנוסח המבחן (ותו לא) ניתן להפנות לצוות הקורס בשעות הקבלה, שיפורסמו באתר הקורס.
- **לוח הזמנים:** לטובת פתרון מבחן הבית יעמדו לרשותכם כ-72 שעות. תכננו ונהלו את זמנכם היטב. שימו לב שאינכם מבזבזים זמן רב מדי על דברים שאינם מהותיים לתרגיל. כתבו קוד ראשוני ופשוט שעובד ורק לאחר מכן שפרו אותו – עדיף להגיש קוד ראשוני שעובר presubmission מאשר קוד שהושקעה בו מחשבה על חלק קטן מאוד, אך בכללותו אינו מצליח לעמוד בדרישות הבסיסיות של התרגיל. **אל תשכחו לגבות את עבודתכם.**
- **זמינות הסגל לשאלות ופורום בעיות טכניות:** במבחן הבית, בשונה מתרגילי הבית שהגשתם במהלך הקורס, לא יפתח פורום לשאלות טכניות או מהותיות. במקום זאת, נקיים מדי יום שעות קבלה מרוכזות. בשעות אלה תוכלו לשאול שאלות לגבי **עניינים טכניים בלבד** הקשורים למבחן (לדוגמה שאלות לגבי חוסר בהירות או ניסוח). **שאלות אחרות לא יענו (וחבל על הניסיון והזמן של כולנו...).** השאלות הנפוצות שישאלו בשעות הקבלה יפורסמו במרכז באשכול מתעדכן, בחלק של המבחן באתר הקורס. **עקבו אחריו.**
- **פתרון בית הספר:** למבחן הבית לא יסופק פתרון בית הספר. באתר הקורס מצורפות דוגמאות מינימליות של קלט/פלט ו/או דוגמא מינימלית של תוכנית העושה שימוש במחלקה שתדרשו לממש. הדוגמה מכילה תיעוד רב מאוד ומטרתה לסייע לכם בהבנת התוכנית. קראו אותה.
- **Pre-Submission:** במבחן זה ה-pre-submission script יהיה מנימלי במיוחד. מטרתו היא לוודא שקובץ ה-tar שהגשתם אכן מכיל את הקבצים הנחוצים וכי הקובץ שהגשתם אכן עובר הידור בהצלחה. לכן, יתכן ולא נבצע בדיקות תקינות כלל על הקובץ שהגשתם (ואם כן, הן יהיו מינימליות ביותר). לכן, שימו לב שאינכם מסתמכים על תוצאת ה-presubmit בשום שלב, שכן איננה מהווה ביטחון לכך שהקוד תקין ולא נקבל ערעורים שבסיסים יהיה צליחת ה-presubmission בלבד.

- **הארכות זמן:** בניגוד לתרגילי הבית, **לא תינתן ארכה להגשת מבחן הבית בגין כל סיבה שהיא**. כשקבענו את רמת הקושי של המבחן מצד אחד, ואת כמות הזמן שהוקצתה לפתרונו אותו מצד שני, לקחנו בחשבון נסיבות אישיות, לחץ זמן בתקופת המבחנים וכל סיבה נוספת שעלולה לעלות. לכן, בפרט, גם זכאות לתוספת זמן במבחנים פרונטלים אינה מזכה בארכה להגשת מבחן זה. כמו כן, לא ניתן "לצבור" ארכה בין המבחנים או על חשבון בנק ימי החסד שהקצנו להגשת התרגילים בקורס.

- **אחסון הקוד במקומות חיצוניים:** אם אתם מאחסנים את פתרון המבחן שלכם בשיטת איחסון חיצונית – **מחובתכם לוודא** שאין לאף אדם אחר הרשאה לגשת למבחן. כך למשל, אם אתם משתמשים ב-git (על ידי פלטפורמות כדוגמת github, bitbucket וכו'), עליכם לוודא כי ה-repository שלכם פרטי כך שלאף אדם מלבדכם אין גישה אליו. באופן דומה, אם התיקיה שבה שמור המבחן מגובת אוטומטית באמצעות Dropbox, Google Drive וכו' – עליכם לוודא שלאף אדם אחר אין גישה לתיקיה זו. כך, **גם אם אתם פותרים את המבחן על מחשבי בית הספר, וודאו כי לאף משתמש אחר אין הרשאה לתיקית הפתרון שלכם**.

רשלנות בהגנה על פתרוניכם מהווה עבירת משמעת על תקנון המשמעת האוניברסיטאי. דינם של המעתיק והמועתק – זהה. אנו נבצע בדיקות שגרתיות של אמצעי איחסון חיצוניים על מנת לשמור על טוהר הבחינות.

זכרו. מבחן בית הוא עניין של אמון. היה לנו חשוב לאפשר לכם לקבל ציון מספרי בקורס, כך שיוכל לשקף את מידת ההשקעה שלכם. אנא, אל תגרמו לנו להתעסק עם עבירות משמעת. חבל על הזמן של כולנו

2 רקע

מגפת הקורונה הגיע 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.

מדינת ישראל החליטה לפנות למוסד האקדמי המוביל בישראל - האוניברסיטה העברית על מנת שתפתח תוכנה לעדכן אנשים במהירות על כך שישנו חשש שהם נדבקו 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.

3 התוכנה Spreader Detector

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 תעודת זהות של אדם שזוהה כנשאי קורונה וסרטי וידאו (בהם הוא מזוהה) המתעדים פרק זמן כלשהו בחייו (יכול להיות מספר דקות, מספר שעות ואף ימים).
- (2) מחלצת את המידע מתוך הסרטונים באופן הבא: את מי פגש הנשא, את מי פגשו אלו שהוא פגש וכך הלאה עד שמזהים את כל האנשים שהופיעו בסרטונים ויודעים מי פגש את מי ובאיזה שלב.
- (3) מעבירה את המידע שחילצה למבנה נתונים (פרטים בהמשך),
- (4) מעריכה את הסיכוי של כל אחד מהמועמדים להידבק מהנשא המזוהה.
- (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 הווידאו של אדם שזוהה כנשאי קורונה, תנתחו אותו, תעבדו אותו ותקבעו את סדר עדכון הנדבקים הפוטנציאליים על כך שהם צריכים להיכנס לבידוד או השגחה רפואית או שהם פטורים מכל חובה שהיא.

4 התוכנית SpreaderDetectorBackend

בחלק זה תממשו את החלק העיקרי של התוכנה Spreader Detector. המימוש יעשה במספר שלבים (זה רעיון אפשרי למימוש, לא מחייב. אך במידה ותממשו אחרת תאלצו להסביר את השיקולים בפירוט, יצוין בהמשך כיצד להסביר):

- (1) קבלת הקלט ועיבודו.
 - (2) בניית מבנה הנתונים המתאים.
 - (3) הערכת סיכויי ההדבקה של אדם.
 - (4) איסוף התוצאות ומיון.
 - (5) החזרה פלט המתאר את רשימת האנשים לפי רמת חומרה וכיצד יש לנהוג בכל אחד (פירוט בהמשך).
- *** שימו לב - תצטרכו לפרט בקובץ ה-README וגם להסביר בראיון לאחר המבחן איך בחרתם להתמודד עם כל אחד מהשלבים הללו. טיפ: כתבו את ההסברים תוך כדי העבודה עצמה. הביטו בקובץ readmen המסופק במודל.
- *** שימו לב 2 – המימוש לתוכנית יעשה בקובץ שיקרא [SpreaderDetectorBackend.c](#).

זכרו – התוכנה היא חלק מפרויקט לאומי רחב ונחוץ, כלומר, התוכנה הולכת לקבל קלט גדול מאוד והיא צריכה לפעול מהר וביעילות מיטבית. נתחו את זמני הריצה של הפונקציות שלכם ונסו לשפר אותם ככל הניתן (תשאלו על כך בריאיון).

4.1 קבצים נלווים

[SpreaderDetectorParams.h](#)

הקובץ כולל: את כל הקבועים הרלוונטיים למשימה. ודאו שאתם מבינים כל אחד מהם ואת משמעותו טרם תחילת העבודה. יפורט בהמשך מתי להשתמש בכל אחד מהם. עליכם להשתמש בקבועים אלו.

4.2 קלט

התוכנית תקבל כקלט שני קבצים:

1. [הקובץ People.in](#) - קובץ זה יכלול את הפרטים המזהים של האנשים אשר נקלטו במהלך הסרטונים ("האח הגדול רואה הכל"). מבנה הקובץ יהיה כך:

כל שורה תייצג פרטים מזהים של אדם יחיד. מבנה השורה יהיה:

<Person Name> <Person ID> <Person age>\n

דגשים:

- * 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).
- * 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).
- * גיל - מספר, לא בהכרח שלם, כלשהו בין 0 ל-120 (כולל), כמו בקודמים, ניתן להניח כי הוא תקין.
- * ניתן להניח שאורך שורה מקסימלי הינו 1024 תווים.
- * **חל איסור לקרוא את הקובץ פעמיים לאחר פתיחתו.** קריאת הקובץ פעמיים תוביל להורדת נקודות.

1. הקובץ Meetings.in - קובץ זה יכלול את הפרטים של מפגשים שתועדו בסרטונים. מבנה קובץ זה מעט מורכב יותר, וודאו כי אתם מבינים אותו טרם תיגשו למשימה.
- * **השורה הראשונה** בקובץ תכלול את מספר הזיהוי (person ID) של האדם שזוהה כחולה (ה"spreader"). כלומר, השורה תכלול רצף של 9 ספרות.
- * **השורות הבאות** יחולקו למקבצים, כלומר:
השורות 2 עד k (מספר טבעי כלשהו) יתארו את המפגשים שתועד בהם המדביק.
השורות k+1 עד k+m יתארו מפגשים שתועד בהם האדם הראשון אותו המדביק פגש.
השורות k+m+1 עד k+m+1+n יתארו מפגשים שתועד בהם האדם השני אותו המדביק פגש.
וכך הלאה.
מבנה כל אחת מהשורות הללו:

<infector_id> <infected_id> <distance> <time>

כאשר המרחק הוא מספר, לאו דווקא שלם, ביחידות של מטרים והזמן הוא מספר, לאו דווקא שלם, המתאר את משך הפגישה בדקות.

דוגמא:

```
123456789 // infector id
123456789 234567891 4 10
123456789 345678912 2 16.4
234567891 456789123 7.6 10.2
234567891 678912345 4 18
345678912 567891234 3 5
```

הסבר מילולי לדוגמא:

המדביק שלנו יהיה ת.ז 123456789.
הוא (המדביק) ות.ז מספר 234567891 היו במרחק 4 מטרים במשך 10 דקות בסרטון.
בנוסף, הוא (המדביק) ות.ז מספר 345678912 היו במרחק 2 מטרים במשך 16.4 דקות בסרטון.
ת.ז 234567891 שפגש את המדביק, היו בקרבת ת.ז מס' 456789123 במרחק 7.6 מטרים במשך 10.2 דקות בהמשך הסרטון. והוא גם היה בקרבת ת.ז מס' 678912345 במרחק 4 מטרים במשך 18 דק'.
ידוע לנו גם ש- ת.ז מס' 345678912 שהה בקרבת ת.ז מס' 567891234 במרחק 3 מטרים במשך 5 דקות.

דגשים:

- * מספרי תעודות הזהות, חוקיים ותקינים ויופיעו בהכרח בקובץ People.in.
- * כל המספרים יהיו מטיפוס float למעט תעודות הזהות. המספרים יהיו חיוביים (גדולים או שווים ל-0) כאשר גבולותיהם יוגדרו בקובץ הקבועים הנלווה לחלק זה (הגבולות יהיו קטנים מהערך המקסימלי של משתנה מסוג float). אין לכם להסתמך על הקבועים הנתונים בשום שלב של הפתרון שכן אנו יכולים לשנות אותם בהתאם לאילוצי ורצונות הממשלה (וצוות הקורס, שימו לב לדמיון: גם אצלנו יש שתי "ראשות ממשלה" ומספר המתרגלים בקורס קרוב למספר השרים בממשלה! הפסקה עם ההומור, ממשיכים). כלומר, השתמשו בקבועים ולא בערכיהם!
- * דוגמא לקבצי קלט קיימת במודל.
- * ניתן להניח שאורך שורה מקסימלי הינו 1024 תווים.
- * 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 כגרף קשיר ללא מעגלים ובנוסף, אין קודקוד שמצביעה עליו יותר מקשת אחת. למשל: בדוגמא למעלה, לא ייתכן שת.ז 345678912 יצביע גם על 678912345.
- * **חל איסור לקרוא את הקובץ פעמיים לאחר פתיחתו.** קריאת הקובץ פעמיים תוביל להורדת נקודות.

4.3 פלט

עליכם לכתוב את הפלט של התוכנית שלכם לקובץ SpreaderDetectorAnalysis.out. מבנה הקובץ יהיה כך שבכל שורה יהיו פרטים של אדם יחיד לפי המתואר בסעיף 4.6 (בהמשך).
*** דוגמא לקובץ פלט קיימת במודל.

4.4 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 $time$ (דקות), ההסתברות שידביקו אחד את השני הוא:

$$Crna_{p1,p2}(dist, time) = \frac{time * min_{distance}}{dist * max_{time}}$$

שימו לב - הפונקציה תחזיר תמיד מספר בין 0 ל-1 (הפרמטר $dist$ תמיד שונה מ-0 שכן לא יכולים להיות שני אנשים במרחק 0 מטר אחד מהשני) וזה יהיה הסיכוי של אדם להידבק לפי הנוסחה.

שימו לב 2 - המשתנים $MIN_DISTANCE$ ו- MAX_TIME הינם שני קבועים הנמצאים בקובץ הקבועים המצורף למשימה. מבחינת אינטואיציה, $MIN_DISTANCE$ משקף את המרחק הקטן ביותר אותו ניתן לתפוס בסרטון, המשתנה MAX_TIME הוא משך הזמן הארוך ביותר בו שני אנשים יכולים להיות אחד בקרבת השני והוא יהיה גם משך הסרטון אותו אנחנו מנתחים.

נבחין כי:

$$\lim_{\substack{d \rightarrow min-distance \\ t \rightarrow max-time}} Crna_{p1,p2}(d, t) \rightarrow 1$$

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

$$\lim_{\substack{d \rightarrow \infty \\ t \rightarrow 0}} Crna_{p1,p2}(d, t) \rightarrow 0$$

כלומר, יותר מרחק ופחות זמן אחד עם השני מוריד את סיכויי ההדבקה.

ריחוק חברתי!!! (במיוחד בזמן המבחן, כדי שהקוד שאתם כותבים לא ידבק בדמיון מהקוד של החבר/ה ליד).

4.5 האלגוריתם למציאת ההסתברות להידבקות - פסאודו קוד בסיסי

```
results = []
for person in persons
    get chance the person has been infected
    results.append(person)
sort(results)
let them know by the order of severity
```

4.5.1 חישוב הסיכוי להידבקות

אופן החישוב של סיכויי ההידבקות של אדם צריך להיות באופן הבא:

1. אם אדם נתון היה במפגש עם הנדבק (ה- $spreader$) - 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. אם אנחנו מדברים על אדם במעגל שני - 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 $P2$ היו במגע לאחר ש- $P2$ 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. המקרה הכללי - אם $P1$ הוא חולה מאומת, $P2$ היה במגע עם $P1$, וידוע ש- $P3$ היה במגע עם $P2$, ... , וידוע ש- P_n היה במגע עם P_{n-1} אז הסיכוי של P_n להידבק יהיה:

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

*** אופן החישוב אכן נראה מעט מורכב אך שימו לב כי שימוש באלגוריתם שמימשותם בתחילת התרגיל יכול לסייע כאן (זה רמז עבה מאוד).

** תיקון - בפרמטרים האחרונים זה d_{n-1}, t_{n-1} .

שימו לב ענק – רוב החישובים בתרגיל נעשים עם float, מזכיר לכם שאלו משתנים בין מערכות הפעלה ומחשבים שונים, לשם האחידות, בחנו את התוצאות שלכם על משבי האקווריום בלבד. כלומר, כתבו את כל הקוד שלכם כרגיל על המחשב האישי וכשאתם רוצים לוודא אם הפיתרון שלכם זהה לקבצים שסיפקנו לכם, הריצו את הקובץ שלכם על מחשבי האקווריום ובצעו diff. הקבצים שסיפקנו לכם חושבו על מחשבי האקווריום.

4.6 מיון והתראה לדבקים פוטנציאליים

לאחר כל התהליך, צריכה להיות בידיכם רשימה של הדבקים הפוטנציאליים שזוהו בסרטונים כאשר ידוע מה הסיכוי שכל אחד מהם נדבק (לפי החישובים שהוצגו לעיל), עליכם למיון את הרשימה על פי מידת הדחיפות ולהודיע להם. עליכם להעביר את הפענוח הרפואי למשרד הבריאות **ביעילות הטובה ביותר** (אתם תידרשו להסביר כיצד פעלתם בראיון).

4.6.1 קביעת מידת הדחיפות

1. אם האדם שנמצא הוא בעל סבירות גבוהה מהקבוע (גדול/שווה) MEDICAL_SUPERVISION_THRESHOLD - הוא מוגדר כבעל צורך בהשגחה רפואית.
2. אם האדם שנמצא הוא בעל סבירות גבוהה מהקבוע (גדול/שווה) REGULAR_QUARANTINE_THRESHOLD - הוא מוגדר כבעל צורך בבידוד.
3. אם האדם שנמצא הוא עם סיכון נמוך מהקבוע הקודם, כלומר, הסיכון שלו מינימלי לחלוטין ואין טעם להכניס אותו לבידוד או לחשוד בו – ולכן יוגדר כאדם נקי.

4.6.2 מיון האנשים

המיון יתבצע כך:
ראשית כל אלו שצריכים לקבל השגחה רפואית (Hospitalization), ואת הסדר הפנימי שלהם נקבע ע"פ סיכוי ההידבקות שלהם כאשר הגבוה ביותר יופי בשורה קודמת יותר (שורה גבוהה יותר במסמך הפלט).
לאחר מכן, יופיעו אלו שצריכים להיכנס לבידוד (Quarantine), וגם אותם נשווה בסדר הפנימי ע"פ סיכוי ההידבקות. ולאחריהם, אלו שיצאו תקינים לחלוטין (No serious chance for infection) וגם אותם נמיון לפי סיכוי ההדבקה.

4.6.3 ההדפסה לקובץ הפלט

ההודעה לקובץ הפלט תודפס בפורמט הבא:

1. אם האדם שנמצא הוא בעל סבירות גבוהה מהקבוע (גדול/שווה) REGULAR_QUARANTINE_THRESHOLD עליכם לכתוב לקובץ את השורה הבאה:
"14-days-Quarantine: <name> <id>.\n"
 2. אם האדם שנמצא הוא בעל סבירות גבוהה מהקבוע (גדול/שווה) MEDICAL_SUPERVISION_THRESHOLD (יהיה תמיד גדול יותר מהקבוע הקודם), עליכם לכתוב לקובץ את השורה הבאה:
"Hospitalization Required: <name> <id>.\n"
 3. אם האדם שנמצא הוא עם סיכון נמוך מהקבוע הקודם, כלומר, הסיכון שלו מינימלי לחלוטין ואין טעם להכניס אותו לבידוד או לחשוד בו, עליכם לכתוב לקובץ את השורה הבאה:
"No serious chance for infection: <name> <id>.\n"
- עליכם להדפיס את כל האנשים שהתקבלו בקובץ הקלט המתאר את האנשים, בין אם הם דחופים לטיפול/בידוד/נקיים.
שימו לב – בקובץ הקבועים שסופק לכם כל ההודעות כבר מופיעות, השתמשו בהן ורק בהן.
שימו לב 2 – המלצה לשלב הדיבוג, הדפיסו גם את סיכוי ההדבקה שחישבתם וודאו שהוא תקין.

4.7 הרצת התוכנית

התוכנית תרוץ עם הפקודה הבאה:

```
$ ./SpreaderDetectorBackend <Path to People.in> <Path to Meetings.in>
```

הבהרות:

1. עליכם לבדוק שהקלט לתוכנית אכן תקין (מס' ארגומנטים, קבצים תקינים וכו'):
אם פורמט הקלט לא תקין הדפיסו לstderr (וצאו עם קוד שגיאה מתאים) את ההודעה הבאה:
"Usage: ./SpreaderDetectorBackend <Path to People.in> <Path to Meetings.in>\n"
אם ישנה תקלה בפתיחת אחד הקבצים הדפיסו לstderr (וצאו עם קוד שגיאה מתאים) (למשל - קובץ לא קיים):
"Error in input files.\n"
2. במידה וישנה תקלה בפתיחת קובץ הפלט, הדפיסו לstderr את ההודעה הבאה וצאו עם קוד שגיאה מתאים:
"Error in output file.\n"
3. במקרה של שגיאה בכל פונקציה אחרת מהספרייה הסטנדרטית, הדפיסו לstderr את ההודעה הבאה וצאו עם קוד שגיאה מתאים:
"Standard library error.\n"

עליכם לוודא כי הזיכרון שוחרר במלואו בעת היציאה מהתוכנית בכל מצב.

5 קובץ README

מצורף לבחינה קובץ README בו תדרשו לפרט מידע לגבי המימוש של שלבים בחלק השני. עליכם לתאר את השלבים אליהם אתם נדרשים במימוש חלק ב' של המשימה ב-3-5 משפטים לכל חלק. שימו לב - אתם תשאלו על הדברים שכתבתם שם. ההמלצה היא למלא את זה בזמן אמת. **ניתן לכתוב בעברית.**

6 הצהרת טוהר בחינות

במבחן זה בניגוד לשאר הקורס נדרוש מכם להגיש בנוסף לקבצים רגילים גם קובץ STATEMENT.pdf הכולל הצהרת טוהר בחינות וחתום לידיכם. הקובץ מצורף במודל בחלק של הבחינה.

7 הגשת הבחינה

עליכם להגיש קובץ tar בשם `exam.tar` והוא יכלול את הקבצים הבאים **והם בלבד**:

STATEMENT.pdf
(no suffix!) README
SpreaderDetectorBackend.c

8 הערות ודגשים

- כתבו את כל ההודעות שבהוראות התרגיל בעצמכם. העתקת ההודעות מהקובץ יכולה להוסיף תווים מיותרים לפגוע בבדיקה האוטומטית, המנקדת את עבודתכם.
- אנא וודאו כי התרגיל שלכם עובר את סקריפט ה-pre-submission ללא שגיאות או אזהרות. הסקריפט זמין בנתיב:
`~labcc2/www/c_exam/presubmit`
- 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> -o SpreaderDetectorBackend`
- כחלק מהבדיקה האוטומטית תבדקו על סגנון כתיבת קוד. תוכלו להריץ בעצמכם בדיקה אוטומטית לסגנון הקוד בעזרת הפקודה:
`~labcc2/www/codingStyleCheck <code_files>`
- 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, valgrind 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 שנתנו במשך הקורס - אורך פונקציות, שמות משתנים ופונקציות 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' - דוגמה מלאה

1. בהינתן הקלט הבא:

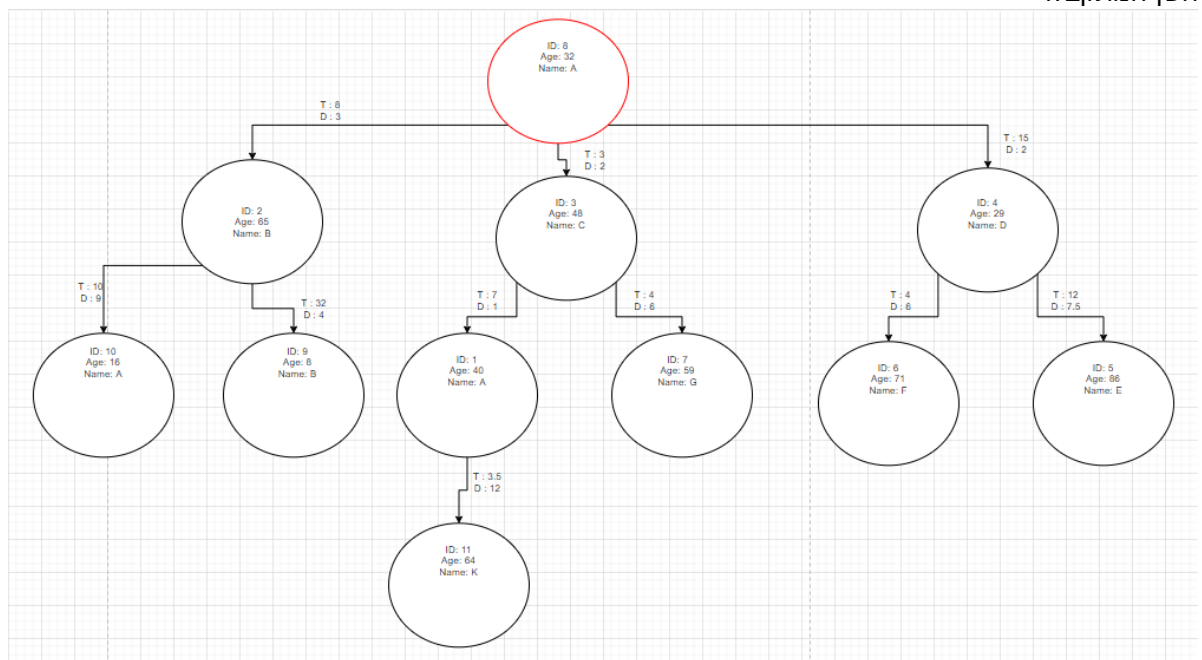
People.in	Meetings.in
A 8 32 B 2 65 C 3 48 D 4 29 A 10 16 B 9 8 A 40 1 G 7 59 F 6 71 E 5 86 K 11 64	8 8 2 3 2 8 3 2 3 8 4 2 15 2 10 9 10 2 9 4 32 3 1 1 7 3 7 6 4 4 6 6 4 4 5 7.5 12 1 11 12 3.5

שימו לב - הID בדוגמא זו הם קצרים וקלים בכוונה כדי שהדוגמא לא תהיה מורכבת ועמוסה מדי. נשתמש כאן בכוונה במספרים קצרים ושמות בסיסיים. קבצים לדוגמה ניתן לראות במודל.

2. נרצה לבנות מבנה נתונים שיכיל את כל המידע הנ"ל.

מטעמי נוחות, נייצג זאת בדוגמא בעזרת עץ כאשר המדבוק (ID מס' 8 בדוגמא, ניתן לראות זאת בשורה הראשונה בקובץ Meetings.in) הוא השורש, האנשים שבאו איתו במגע ראשוני יהיו בניו, ואלו שבאו איתם יהיה הבנים שלהם וכך הלאה.

העץ המתקבל:



*שימו לב: אין חובה להשתמש בעץ, זה מוסבר כך לשם נוחות. עם זאת, זה בהחלט מבנה נתונים אפשרי לשימוש (ישנם עוד כמה אפשריים).

3. נעת נרצה לחשב את הסיכוי של כל אחד מהאנשים להידבק בעזרת הפונקציה Crna ואופן החישוב שהצגנו קודם לכן (סעיף 4.5.1).

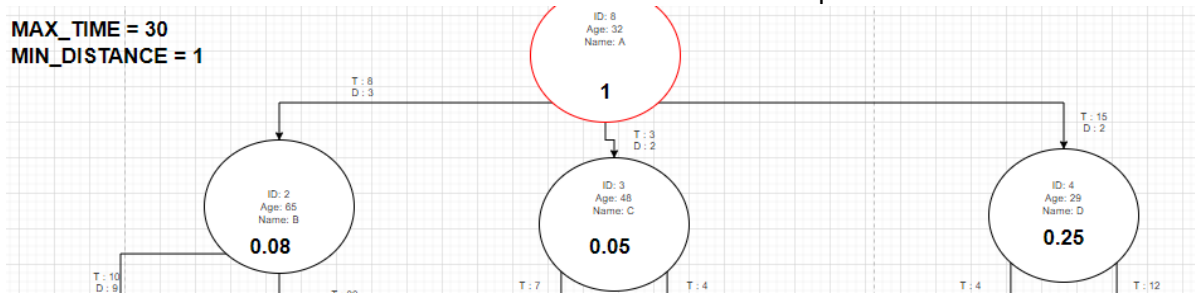
בהינתן הקבועים הבאים:

```
#define MAX_TIME 30.0
#define MIN_DISTANCE 1.0
```

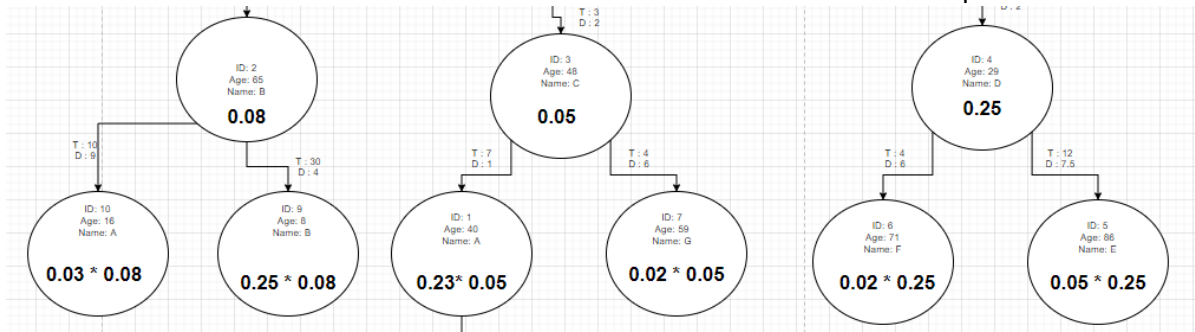
שמתארים לנו צילום סרטונים באורך 30 דקות בשטח כשהמרחק המינימלי שניתן להיות בו 1 מטר. התוצאה המתקבלת (הסיכוי של כל אחד כתוב בתוך העיגול):

עבור הרמה הראשונה בה החישוב קל ומייד:

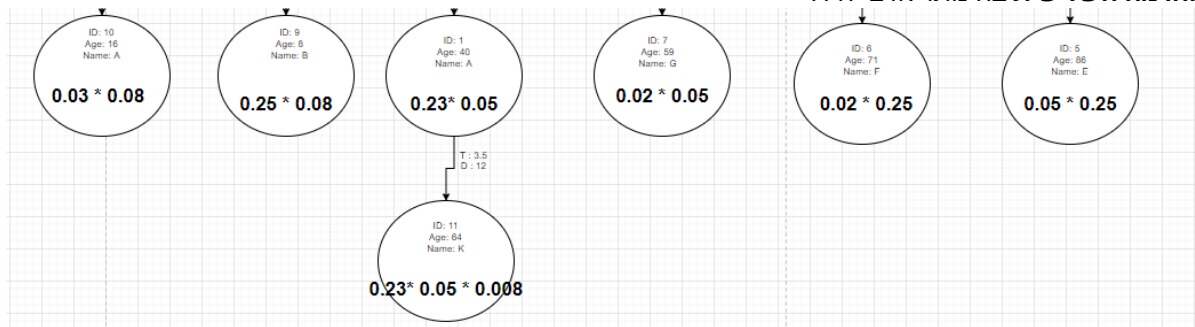
MAX_TIME = 30
MIN_DISTANCE = 1



הרמה השנייה תסתמך על תוצאות הרמה הראשונה:



והרמה השלישית בה נותר אדם יחיד:



4. איגוד התוצאות של האנשים:

ID	AGE	SICK
8	32	1
2	65	0.08
3	48	0.05
4	29	0.25
10	16	0.0024
9	8	0.02
1	40	0.0115
7	59	0.001
6	71	0.005
5	86	0.0125
11	64	0.000092

5. מיון התוצאות לפי סעיף 4.6.

6. הדפסת קובץ הפלט עם ההודעות המתאימות לפי סעיף 4.7
נניח כי הקבועים היו:

```
#define RISK_AGE 65  
#define REGULAR_QUARANTINE_THRESHOLD 0.1  
#define MEDICAL_SUPERVISION_THRESHOLD 0.3
```

7. בניית קובץ הפלט וכתיבה אליו.

דוגמאות נוספות – במודל.