האוניברסיטה העברית בירושלים

בית הספר להנדסה ולמדעי המחשב ע"ש רחל וסלים בנין

סדנת תכנות בשפת C ו־C++ (67315)

תרגיל 5 - שפת C++

תאריך הגשה: 30 ליוני, 2020, בשעה 23:25

נושאי התרגיל: שימוש במבני נתונים בספריית STL ובאלגוריתמים של STL.

1 רקע

לאחר שרווחיו של בית הקולנוע "סינמה סיטי" צנחו בעשרה אחוזים, מנכ"ל בית הקולנוע החליט לגייס צוות של אנליסטים על מנת לחקור מהיכן נובעת הירידה הזאת.

לאחר מחקר מעמיק בנתוני החברה ובנתונים של השוק הם הגיעו למסקנה חד משמעית - קולנוע "יס פלאנט" השתלט על השוק.

צוות האנליסטים חוששים שההשתלטות של יס פלאנט על השוק נובעת ממערכת ההמלצה המשובחת שפיתחו, הממליצה באופן מדויק סרטים ללקוחותיה.

לאחר שהצוות הציג את הנתונים האלה למנכ"ל, הוא החליט לגייס אותך לצורך פיתוח מערכת המלצה דומה לזו של "יס פלאנט", בתקווה שמהלך זה יגדיל את רווחיה של "סינמה סיטי" ותמנע פשיטת רגל.

2 הגדרות

1. **נורמה**

נורמה היא פונקציה ממשית המוגדרת על מרחב וקטורי, ומתאימה לכל וקטור ערך ממשי, באופן שמקיימות האקסיומות הבאות:

(א) חיוביות:

$$||x|| = 0 \implies x = 0 \ \wedge \ ||x|| \geq 0$$

(ב) הומוגניות:

$$||\lambda x|| = ||\lambda|| \cdot ||x||$$

(ג) אי שיוויון המשולש:

$$||x|| + ||y|| \geq ||x + y||$$

2. **מכפלה סקלרית**

מכפלה סקלרית היא פעולה על שני וקטורים מהמרחב האוקלידי $\mathbb{R}^n$, שמחזירה סקלר. לדוגמא, יהיו $\alpha, \beta \in \mathbb{R}^n$, המוגדרים באופן הבא:

$$\alpha = (\alpha_1, ..., \alpha_n)$$

$$\beta = (\beta_1, ..., \beta_n)$$

אזי המכפלה הסקלרית בין α, β הינה מוגדרת ומסומנת:

$$\alpha \cdot \beta = \alpha_1 \cdot \beta_1 + ... + \alpha_n \cdot \beta_n$$

3. **הנורמה הסטנדרטית במרחב האוקלידי**

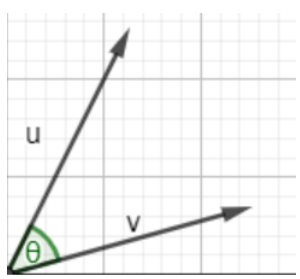
$$||x|| = \sqrt{x_1^2 + ... + x_n^2}$$

זוהי הנורמה בה נשתמש לאורך כל התרגיל.

4. **זווית בין וקטורים**

בהינתן וקטורים u, v, נוכל לחשב את הזווית θ בניהם בצורה הבאה:

$$\theta = cos^{-1}(\frac{u \cdot v}{||u|| \cdot ||v||})$$



איור 1: זווית בין וקטורים

הערה: הפונקציה ההופכית של cos מוגדרת בתחום $[-1, 1]$ ומונוטונית יורדת בתחום זה, משמע ככל ש־$\frac{u \cdot v}{||u|| \cdot ||v||}$ מתקרב ל־1, הזווית ביןu וv קטנה, וככל ש$\frac{u \cdot v}{||u|| \cdot ||v||}$ מתקרב ל־(1־), הזווית גדלה.

מסקנה: נוכל למדוד דמיון בין ווקטורים (דמיון הכיוונים) לפי חישוב הזווית $\theta = \frac{u \cdot v}{||u|| \cdot ||v||}$. ככל שערך זה יותר גבוה, הווקטוריםu וv דומים יותר.

שימו לב כי אם $1 = \frac{u \cdot v}{||u|| \cdot ||v||}$, למעשה $u = v$.

3 בניית מערכת המלצות

בתרגיל זה תידרשו לבנות מערכת המלצות שתנצח את זו של "יס פלאנט", ובכך להחזיר את רשת הקולנוע "סינמה סיטי" לשוק.

3.1 קלט

מערכת ההמלצה מקבלת כקלט שני קבצי טקסט.

3.1.1 קובץ המכיל מידע על הסרטים לפי תכונותיהם.

	Scary	Funny	Dramatic	Surprising
Twilight	3	4	5	6
Titanic	7	2	9	1
Batman	2	6	4	8
ForestGump	1	7	7	6
StarWars	3	3	4	9

טבלה 1: מבנה קובץ הקלט הראשון

- עבור כל סרט, ולכל תכונה, יש ברשותנו $score$ המייצג כמה התכונה תואמת את הסרט. ה$scores$ הם מספרים שלמים חיוביים מ1 עד 10 (כולל).
- הטבלה היא מלאה - כלומר כל הסרטים הנתונים דורגו עבור כל תכונה נתונה.
- ניתן להניח כי אין רווחים בשם הסרט.
- לצורך פשטות, בקבצים נתעלם מההשורה הראשונה בה מוצגים שמות של התכונות הנחקרות ונתייחס רק לערכים הרלוונטים.

3.1.2 קובץ המכיל דירוגים של סרטים לפי שמות משתמשים

קובץ זה מייצג את הדירוגים של הלקוחות של רשת "סינמה סיטי" - עבור סרטים שהם ראו (בהנחה כי בכל סיום של צפייה בסרט התבקשו לדרג את הסרט לפי מספר מ1 עד 10 (או ערך ריק (NA)) במידה ולא ראו את הסרט).

	Titanic	Twilight	ForestGump	Batman	StarWars
Sofia	4	NA	8	NA	NA
Michael	NA	8	4	NA	9
Nicole	NA	NA	5	2	6
Arik	NA	8	NA	3	NA

טבלה 2: מבנה קובץ הקלט השני

ניתן להניח כי:

1. לא קיימים שני לקוחות בעלי אותו שם.
2. לא קיימים שני סרטים בעלי אותו שם.
3. כל משתמש דירג לפחות סרט אחד.
4. כל משתמש לא דירג לפחות סרט אחד.

שימו לב: סדר הסרטים בעמודות בקובץ זה לא בהכרח תואם את סדר השורות בקובץ מלעיל.

3.2 אלגוריתמים

3.2.1 המלצה לפי תוכן

רעיון כללי - נמליץ סרטים ללקוח x לפי סרטים שדומים למה שלקוח x דירג גבוה. נרצה להמליץ ללקוח x על סרט שאנו מאמינים שיאהב.

- **שלב 1:** נחסיר את ממוצע הדירוגים של לקוח x מהדירוגים שלו, כדי לנרמל את הדירוגים.
- **שלב 2:** ניצור וקטור העדפה של תכונות ללקוח x המורכב מהדירוגים שלו לסרטים, ביחד עם תכונותיהם של אותם סרטים.

  שימו לב כי וקטור התוצאה משלב 2 בקאורדינטה הi מייצג את המשקל שלקוח x נותן לתכונה הi , כלומר כמה הוא "אוהב" את התכונה הi.
- **שלב 3:** נחשב את הדמיון על ידי חישוב הזווית בין וקטור ההעדפה של לקוח x לבין כל אחד מוקטורי התכונות של הסרטים אותם לקוח x לא דירג - ונמליץ על הסרט עם הדמיון המקסימלי בתכונות.

דוגמא:
נרצה להמליץ ל$Sofia$ על סרט לפי שיטת המלצה לפי תוכן.

- **שלב 1**

  וקטור הדירוגים של סופיה הוא:

	Titanic	Twilight	ForestGump	Batman	StarWars
Sofia	4	NA	8	NA	NA

  - ממוצע הוקטור הינו $\frac{4+8}{2} = 6$.
  - **שימו לב** כי לא התייחסנו לערכים הריקים בחישוב הממוצע.

  נקבל כי וקטור הדירוגים המנורמל של סופיה הוא:

	Titanic	Twilight	ForestGump	Batman	StarWars
Sofia	-2	NA	2	NA	NA

- **שלב 2**

  ניצור את ווקטור העדפה של Sofia.

  1. הדירוג המנורמל של Sofia ל-Titanic הוא 2-, ווקטור התכונות של טיטאניק יהיה:

Titanic	7	2	9	1

  2. הדירוג המנורמל של Sofia לForestGump הוא 2, ווקטור התכונות של ForestGump הוא:

ForestGump	1	7	7	6

  3. נקבל כי בסה"כ וקטור ההעדפות של סופיה הינו:

$$-2 \cdot \begin{pmatrix} 7 \\ 2 \\ 9 \\ 1 \end{pmatrix} + 2 \cdot \begin{pmatrix} 1 \\ 7 \\ 7 \\ 6 \end{pmatrix} = \begin{pmatrix} -12 \\ 10 \\ -4 \\ 10 \end{pmatrix}$$

אינטואיציה: סופיה אוהבת מאוד סרטים מפתיעים ומצחיקים (באותה מידה),לא אוהבת סרטים דרמטיים ומאוד לא אוהבת סרטים מפחידים.

- **שלב 3**

חישוב הדמיון בין וקטור ההעדפות של סופיה לוקטורי התכונות של הסרטים שSofia לא דירגה - Twilight,Batman, StarWars.

1. וקטור התכונות של Twilight : | Twilight | 3 | 4 | 6 | 5 |
כלומר, הדמיון בין התכונות שלו לבין ההעדפות של Sofia הינו:

$$\frac{\begin{pmatrix}-12\\10\\-4\\10\end{pmatrix}\cdot\begin{pmatrix}3\\4\\6\\5\end{pmatrix}}{||\begin{pmatrix}-12\\10\\-4\\10\end{pmatrix}||\cdot||\begin{pmatrix}3\\4\\6\\5\end{pmatrix}||}=\frac{30}{\sqrt{360}\cdot\sqrt{86}}=0.17$$

2. וקטור התכונות של Batman : | Batman | 2 | 6 | 4 | 8 |
כלומר, הדמיון בין התכונות שלו לבין ההעדפות של Sofia הינו:

$$\frac{\begin{pmatrix}-12\\10\\-4\\10\end{pmatrix}\cdot\begin{pmatrix}2\\6\\4\\8\end{pmatrix}}{||\begin{pmatrix}-12\\10\\-4\\10\end{pmatrix}||\cdot||\begin{pmatrix}2\\6\\4\\8\end{pmatrix}||}=\frac{100}{\sqrt{360}\cdot\sqrt{120}}=0.48$$

3. וקטור התכונות של StarWars : | StarWars | 3 | 3 | 4 | 9 |
כלומר, הדמיון בין התכונות שלו לבין ההעדפות של Sofia הינו:

$$\frac{\begin{pmatrix}-12\\10\\-4\\10\end{pmatrix}\cdot\begin{pmatrix}3\\3\\4\\9\end{pmatrix}}{||\begin{pmatrix}-12\\10\\-4\\10\end{pmatrix}||\cdot||\begin{pmatrix}3\\3\\4\\9\end{pmatrix}||}=\frac{68}{\sqrt{360}\cdot\sqrt{115}}=0.33$$

מסקנה: נמליץ לSofia על הסרט Batman מכיוון שוקטור התכונות שלו הכי דומות להעדפות של Sofia!

3.2.2 המלצה לפי סינון שיתופי סרט־סרט

רעיון כללי:
נרצה לתת המלצה ללקוח על סרט שהוא לא ראה, בהסתמך על הסרטים שהוא ראה שהם הדומים ביותר לאותו הסרט שלא ראה.

כלומר, עבור סרט m, נמצא סט $N = \{N_1, .., N_k\}$ המכיל k סרטים שהכי דומים לסרט m <u>וגם</u> שהלקוח דירג.
לפי זה, <u>נחזה $(predict)$ את הדירוג של לקוח x</u> עבור הסרט m.
נעשה זאת באמצעות פונקציה הלוקחת את הדירוגים של הלקוח ה־x עבור אותם $N_1, .., N_k$ הסרטים בסט, ומחשבת את הדימיון (זוויות) בין כל N_j , $j \in [1, k]$ לסרט m.

על מנת לחזות את הדירוג של לקוח x עבור סרט m, נפעל בצורה הבאה:

- **<u>שלב 1:</u>**

  נמצא את הסט $N = \{N_1, .., N_k\}$ של k הסרטים שהכי דומים לסרט m וגם שלקוח x דירג.

- **<u>שלב 2:</u>**

  נחזה את הדירוג של לקוח x לסרט הm באופן הבא:

$$r_{x-m} = \frac{\sum\limits_{j \in N} s_{m-j} \cdot r_{x-j}}{\sum\limits_{j \in N} s_{m-j}}$$

  כאשר s_{m-j} הוא הדמיון בין הסרט הm לסרט הj , וr_{x-m} הוא הדירוג של הלקוח x עבור הסרט m.

כלומר, על מנת להמליץ ללקוח x על סרט, נוכל לחזות את הדירוג שלו עבור כל סרט אותו לא דירג, ולהמליץ לו על הסרט בעל הדירוג הגבוה ביותר שחזינו.

ניקח דוגמא עבור $k = 2$:
נרצה לחזות את הדירוג של Nicole עבור סרט שהיא לא דירגה, ולבסוף להמליץ לה על סרט בעל הדירוג הגבוה ביותר שחזינו.
נעשה את האבחנה הבאה:

- הסרטים אותם Nicole לא דירגה הם: Titanic, Twilight.
- הסרטים אותם Nicole דירגה הם: ForestGump, StarWars, Batman.

כעת נחזה את הדירוג שNicole הייתה נותנת עבור הסרטים אותם לא ראתה ־ על סמך מה שכבר דירגה.

- בשביל לחזות את הדירוג של Nicole עבור Titanic, נמצא את הדימיון בינו לבין הסרטים שראתה ודירגה.
  - הדמיון בין Titanic לForestGump הוא:

$$\frac{\begin{pmatrix} 7 \\ 2 \\ 9 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 7 \\ 7 \\ 8 \end{pmatrix}}{\| \begin{pmatrix} 7 \\ 2 \\ 9 \\ 1 \end{pmatrix} \| \cdot \| \begin{pmatrix} 1 \\ 7 \\ 7 \\ 8 \end{pmatrix} \|} = \frac{92}{\sqrt{135} \cdot \sqrt{163}} = 0.62$$

  - הדמיון בין Titanic לStarWars הוא:

$$\frac{\begin{pmatrix}7\\2\\9\\1\end{pmatrix}\cdot\begin{pmatrix}3\\3\\4\\9\end{pmatrix}}{||\begin{pmatrix}7\\2\\9\\1\end{pmatrix}||\cdot||\begin{pmatrix}3\\3\\4\\9\end{pmatrix}||}=\frac{72}{\sqrt{135}\cdot\sqrt{115}}=0.57$$

- הדמיון בין Titanic לBatman הוא:

$$\frac{\begin{pmatrix}7\\2\\9\\1\end{pmatrix}\cdot\begin{pmatrix}2\\6\\4\\8\end{pmatrix}}{||\begin{pmatrix}7\\2\\9\\1\end{pmatrix}||\cdot||\begin{pmatrix}2\\6\\4\\8\end{pmatrix}||}=\frac{70}{\sqrt{135}\cdot\sqrt{120}}=0.55$$

כעת, ידוע כי $k=2$, ולכן נבחר סט N עם 2 הסרטים שדורגו בצורה הדומה ביותר לTitanic, ובמקרה הזה $N=\{ForestGump, StarWars\}$.

הדירוג של Nicole עבור ForestGump וStarWars הוא 5 ו6 בהתאמה ולכן חיזוי הדירוג של Nicole עבור Titanic הוא:

$$\frac{0.62\cdot 5+0.57\cdot 6}{0.62+0.57}=\frac{6.52}{1.19}=5.478$$

שימו לב שבקלט הדירוגים הם שלמים אך הדירוגים שחזינו יכולים להיות שבריים.

- עבור Twilight נחזור על אותו התהליך בדיוק, ונקבל חיזוי של הדירוג: 3.52.

מסקנה: נמליץ על Titanic מכיוון שהדירוג החזוי של Nicole לסרט Titanic הוא הגבוה ביותר מתוך הדירוגים שחזינו עבור הסרטים בN.

3.3 API

עליכם לממש מחלקה בשם RecommenderSystem עם המתודות הבאות:

1. **loadData(moviesAttributesFilePath, userRanksFilePath)**

מתודה זו תקבל כקלט שתי מחרוזות אשר מכילים את ה$path$ של שני קבצי הקלט ותטען את המידע למערכת ההמלצה שלכם.

הפונקציה תחזיר 0 במידה של הצלחה ו1- במידה של כישלון.

במידה והpath לא תקין, עליכם בנוסף להדפיס הודעות שגיאה "Unable to open file <file_path>".

שימו לב: מאגר המידע של סינמה סיטי איתו אתם עובדים נמצא בשרת רחוק המצריך פעולות תקשורת מרובות על מנת לגשת אליו.

לכן, עליכם להמנע מגישות I/O $(Input/Output)$ מיותרות, ולגשת לקבצים **פעם אחת בלבד** על מנת לקרוא את המידע מתוך הקבצים.

2. **recommendByContent(userName)**

מתודה זו תקבל כקלט מחרוזת המייצגת את שם הלקוח ותחזיר מחרוזת המייצגת את שם הסרט הממולץ לפי אלגוריתם המלצה לפי תוכן.

אם userName לא קיים במערכת, יש להחזיר "USER NOT FOUND" .

3. **predictMovieScoreForUser(movieName, userName, k)**

מתודה זו תקבל שתי מחרוזות המייצגות את השם של הלקוח ואת השם של הסרט עבורם רוצים לחזות את הדירוג, k מספר שלם וחיובי (הפרמטר באלגוריתם הסינון השיתופי) המייצג את מספר הסרטים הדומים ביותר (ומדורגים על ידי userName) ל־movieName, עליהם נתבסס בחיזוי.

אם userName או movieName לא קיימים במערכת, יש להחזיר 1־.

המתודה מחזירה מספר חיובי עשרוני שהינו חיזוי הדירוג של userName עבור movieName לפי שיטת הסינון השיתופי.

- ניתן להניח כי הפרמטר k קטן ממספר הסרטים שדורגו על ידי userName.
- ניתן להניח כי אם userName ו־movieName קיימים במערכת, אזי userName לא דירג את movieName.

4. **recommendByCF(userName, k)**

מתודה זו מקבלת מחרוזת המצייגת את השם של הלקוח עבורו רוצים לתת המלצה,ו־ k מספר שלם וחיובי (הפרמטר באלגוריתם הסינון השיתופי) המייצג את מספר הסרטים הדומים ביותר (ומדורגים על ידי userName), בעזרתם נעשה את החיזוי.

המתודה מחזירה את הסרט עליו נמליץ לuserName לפי שיטת סינון שיתופי כפי שהוסבר מלעיל.

- אם userName לא קיים במערכת, יש להחזיר "USER NOT FOUND" .
- ניתן להניח כי הפרמטר k קטן ממספר הסרטים שדורגו על ידי userName.

5. **שימו לב שהקוד שאתם מגישים אינו מכיל main!**

דוגמא לשימוש במחלקה עם הקלט שבדוגמא:

```
RecommenderSystem rec;
rec.loadData(movieAttributesPath,usersRanksPath);
std::cout<<rec.recommendByContent( userName: "Sofia")<<std::endl;
std::cout<<rec.predictMovieScoreForUser( movieName: "Twilight", userName: "Nicole", k: 2)<<std::endl;
std::cout<<rec.predictMovieScoreForUser( movieName: "Titanic", userName: "Nicole", k: 2)<<std::endl;
```

איור 2: דוגמא לשימוש במחלקה

```
Batman
3.5244
5.46432
```

איור 3: פלט הדוגמא

3.4 הערות

1. לצורך קריאת הנתונים מהקבצים, הנכם רשאים (ומומלץ) להעזר במחלקה istringstream.
2. שימו לב כי באלגוריתם המלצה לפי תוכן אנו משתמשים בדירוגים מנורמלים ובאלגוריתם המלצה לפי סינון שיתופי אנו משתמשים בדירוגים המקוריים. **המנעו מלטעון מידע כפול וחשבו על דרכים אחרות להתגבר על כך.**

3. הפונקציה loadData **לא תיקרא יותר מפעם אחת** על אותו האובייקט.

4. על מנת לבדוק את הקוד שלכם, תוכלו למצוא במודל שני קבצי קלט לדוגמא המכילים את הדוגמא שראיתם במסמך זה בפורמט הנכון.

3.5 בדיקת הקוד

את הקוד שלכם נבדוק כספרייה עם השימוש המתואר למעלה, אבל אנחנו גם מספקים לכם פתרון בית ספר כexecutable על מנת שתוכלו לבדוק את התשובות שלכם (לכל פונקציה שתממשו בנפרד).
איך תוכלו להריץ את הפתרון בית ספר?

1. הרצת הפונקציה recommendByContent(userName):

./RecommenderSystem movie_data_path.txt ranking_data_path.txt content recommend userName

2. הרצת הפונקציה predictMovieScoreForUser(movieName, userName, k):

./RecommenderSystem movie_data_path.txt ranking_data_path.txt cf predict movieName userName k

3. הרצת הפונקציה recommendByCF(userName, k):

./RecommenderSystem movie_data_path.txt ranking_data_path.txt cf recommend userName k

4 נהלי הגשה

- **קראו בקפידה את הוראות תרגיל זה ואת ההנחיות להגשת תרגילים שבאתר הקורס. כמו כן, זכרו כי התרגילים מוגשים ביחידים. אנו רואים העתקות בחומרה רבה!**

- **עליכם להשתמש במבני הנתונים בספריית STL וכן מומלץ להשתמש באלגוריתמים המוצעים בספריה.** מטרת התרגיל היא **שימוש ניכר במבני נתונים של STL וכן באלגוריתמים** - שימו לב כי קוד נכון ויעיל הוא קוד המשתמש במבנים הנכונים והיעילים ביותר למשימה, ומכאן גם קוד המשתמש באלגוריתמים שהספריה מציעה.

- כתבו את כל ההודעות שבהוראות התרגיל בעצמכם. העתקת ההודעות מהקובץ עלולה להוסיף תווים מיותרים ולפגוע בבדיקה האוטומטית, המנקדת את עבודתכם.

- שימו לב להערות ולהנחות שניתנו לכם, ובעיקר לאלו המסומנות באדום! המנעו מאיבוד נקודות מהסיבה שלא שמתם לב להוראות המודגשות.

- אנא וודאו כי התרגיל שלכם עובר את ה-Pre-submission Script **ללא שגיאות או אזהרות**. קובץ ה-Pre-submission Script זמין בנתיב:

~labcc2/www/ex5/presubmit_ex5

- על מנת לבדוק את פתרונכם, הריצו את פיתרון בית הספר על מגוון קלטים אפשריים כפי שהונחתם בסעיף 3.5, בנתיב:

~labcc2/www/ex5/school_solution/RecommenderSystem

שימו לב כי בדיקת הpre-submission אינה מספיקה ועליכם להריץ בעצמכם טסטים לפי התנהגות פיתרון בית הספר!

- עליכם ליצור קובץ tar בשם "**ex5.tar**" (**ובשם זה בלבד**) הכולל את הקובץ **RecommenderSystem.cpp** ו-**RecommenderSystem.h**. שימו לב כי קובץ הheader צריך לכלול את כל ההצהרות של הפונקציות שבניתן במחלקה. ניתן ליצור tar כדרוש על ידי הפקודה הבאה בטרמינל:

$ tar -cvf ex5.tar RecommenderSystem.cpp RecommenderSystem.h

- **שימו לב:** תרגילים שלא הוגשו בקובץ בפורמט tar, או שיוגשו בשם השונה מ־"ex5.tar", לא יבדקו כלל ויקבלו ציון 0. נושא זה לא נבדק ב־Pre-Submission Script.

- שימו לב כי בקבצי התרגיל אתם לא מגישים פונקציית main, אלא **רק את המחלקה**, אך עליכם לבדוק כי התוכנית מתקמפלת כאשר אתם מכניסים פונקציית main המדמה הרצה של class כמו בדוגמא שניתנה לכם, ולפי הפקודה הבאה:

g++ -Wall -Wvla -Wextra -Werror -g -std=c++17 <code files> -o prog

- כחלק מהבדיקה האוטומטית תיבדקו על סגנון כתיבת קוד. תוכלו להריץ בעצמכם בדיקה אוטומטית לסגנון הקוד בעזרת הפקודה:

$ ~labcc2/www/codingStyleCheck <code file or directory>

כאשר <directory or file> מוחלף בשם הקובץ אותו אתם רוצים לבדוק או בשם התיקייה שתרצו לבדוק את כל הקבצים שנמצאים בה.

- במידה והשתמשתם בהקצאות זיכרון, עליכם לדאוג לניהול ושחרור הזיכרון ללא דליפות. תוכלו להיעזר ב־valgrind כדי לבדוק האם בתרגילכם יש דליפות זיכרון. עליכם להריץ את הפקודה:

valgrind --leak-check=full <Command to Debug>

בהצלחה!