

## תרגיל בית 4 - ++C בסיסי

מועד ההגשה:	יום ד' 13/06/18
האחראי על התרגיל:	יוחאי ריעאני
syohai@campus	

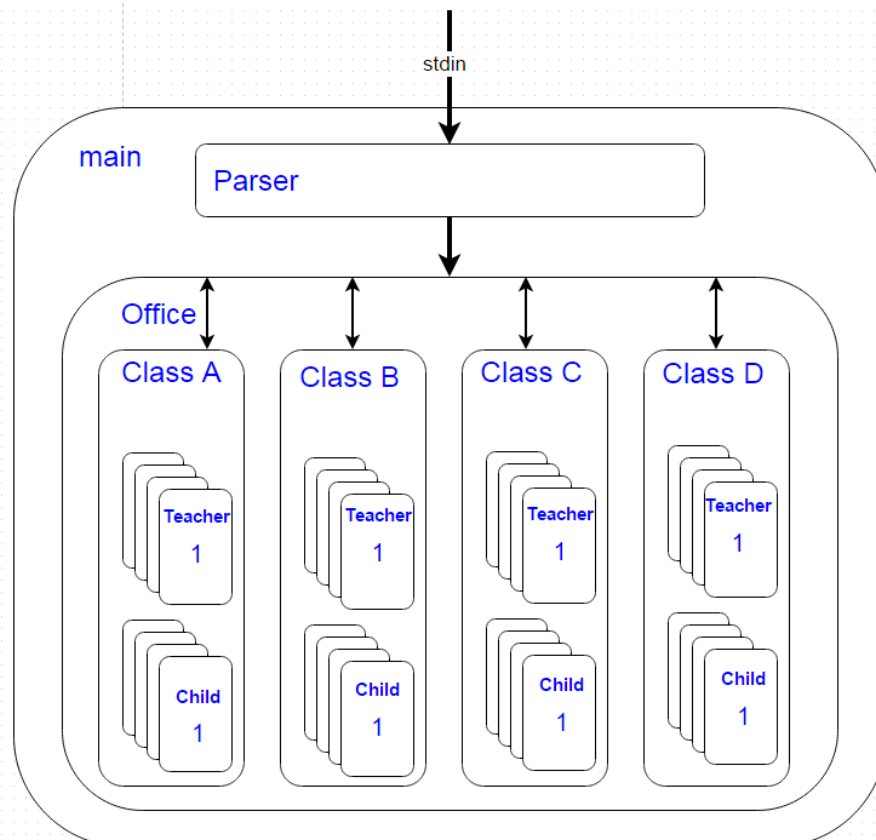
### גן ילדים

בתרגיל זה נבנה גן ילדים. בגן ישנו משרד, כיתות, ילדים וגננות. כמו בכל גן, הכיתות מחולקות לפי גיל הילדים ובכל כיתה מוגדר יחס מקסימלי של מספר ילדים לגננת אחת. בנוסף, ילדים לפעמים חולים ובמצב זה הגן עושה הכל כדי להתקשר להורי הילד. עד שהם לוקחים אותו הגן עסוק ולא מקבל פניות מכל סוג(ראו בהמשך).

חלוקה נוחה של המימוש נתונה להלן:

1. מחלקות אדם, ילד וגננת – כמובן שכל גננת וכל ילד היא/הוא אדם.
2. מחלקות חדר, כיתה ומשרד – כל כיתה ומשרד הם בראש ובראשונה חדר. בנוסף, המשרד מכיל רשימה של כל הכיתות וכיתה מכילה רשימה של ילדים וגננות.
3. ניתוב קלט ופלט – ניתוח הקלט שמוכנס על ידי המשתמש (parsing) וקריאה מתאימה לפונקציות.

להלן, תיאור גרפי של מבנה התוכנית:



שימו לב כי נתון לכם הקובץ defs.H המכיל את ההגדרה הבאה:

```
enum Result {FAILURE, SUCCESS};
```

במהלך התרגיל, במקרים בהם מתודת מחזירה הצלחה או כישלון, יש להשתמש בטיפוס זה.

## חלק 1 – מחלקות אדם, ילד וגננת

### מחלקת אדם

בדומה למה שהוגדר בתרגול יש לממש מחלקה בשם class Person . מחלקה זו תמומש בקבצים . Person.H, Person.C

### משתני מצב של אדם:

1. שם – אובייקט מטיפוס מחרוזת. מומלץ להשתמש במחלקה string (ולא בטיפוס char\*) - לשם כך, יש לבצע include לספרייה <string>.
2. גיל – מספר שלם חיובי.

### הפעולות שנרצה לבצע על אדם:

1. יצירת אדם – נדרש לקבל שם וגיל
2. פעולות get :
1. קבלת גיל – החזרת גילו של האדם
2. קבלת שם – החזרת שמו
3. הדפסה – הדפסת כל משתני המצב של האדם בפורמט הבא לדוגמא:

*Name : Methuselah Ben-Enoch*

*Age : 969*

**שימו לב:** ישנו רווח בודד לפני ולאחר הנקודתיים. פרט זה יחזור על עצמו בכל ההדפסות של תרגיל זה! לנוחותכם, הקבצים Person.H, Person.C מכילים שלד של המימוש.

### מחלקת ילד

יש לממש מחלקה בשם class Child . מחלקה זו תמומש בקבצים Child.H, Child.C

### משתני מצב של ילד:

1. כל משתני המצב של אדם ובנוסף:
2. טלפון – אובייקט מטיפוס מחרוזת שמחזיק את מספר הטלפון של הורי הילד. גם כאן מומלץ להשתמש במחלקה string
3. ילד חולה – דגל בוליאני שמחזיר אמת אם ורק אם הילד חולה .

### הפעולות שנרצה לבצע על ילד:

1. יצירת ילד – נדרש לקבל שם, גיל ומחרוזת שמכילה את מספר הטלפון של ההורים. כברירת מחדל, ילד שנוצר הוא בריא.
2. פעולות get :
1. קבלת טלפון – החזרת מחרוזת שמכילה את מספר הטלפון של ההורים.
2. פעולות set :
1. שינוי טלפון – שינוי מספר הטלפון של ההורים.
2. קביעת הילד כחולה – אם הילד בריא אז נסמן אותו כחולה ונחזיר הצלחה. אחרת, נחזיר כישלון.
4. הדפסה – הדפסת כל משתני המצב של הילד בפורמט הבא לדוגמא:

*Name : Dani Din*

*Age : 2*

*Parent's Phone : 0555123456*

*Is sick : false*

הערה: רצוי להשתמש ב std::boolalpha . ניתן לקרוא על כך באינטרנט.

הטכניון - מכון טכנולוגי לישראל  
הפקולטה להנדסת חשמל  
מבוא למערכות תוכנה

**מחלקת גננת**

יש לממש מחלקה בשם class Teacher . מחלקה זו תמומש בקבצים Teacher.H, Teacher.C .

**משתני מצב של גננת:**

1. כל משתני המצב של אדם ובנוסף:
2. שנות ותק – שלם חיובי שמחזיק את שנות הותק של הגננת.

**הפעולות שנרצה לבצע על גננת:**

1. יצירת גננת – נדרש לקבל שם, גיל ושנות ותק.
2. פעולות get :
1. קבלת הותק – מספר שנות הותק של הגננת.
3. הדפסה – הדפסת כל משתני המצב של הגננת בפורמט הבא לדוגמא:

*Name : Miriam Roth*

*Age : 50*

*Seniority : 10*

## חלק 2 - מחלקות חדר, כיתה ומשרד

### מחלקת חדר

יש לממש מחלקה בשם class Room . מחלקה זו תמומש בקבצים Room.H, Room.C .

#### משתני מצב של חדר:

1. מספר החדר – שלם חיובי.
2. גודל החדר במטר מרובע – שלם חיובי.
3. חדר מאוכלס – דגל בוליאני שמחזיר אמת אם ורק אם החדר מאוכלס.

#### הפעולות שנרצה לבצע על חדר:

1. יצירת חדר – נדרש לקבל מספר וגודל.
2. פעולות get :
  1. קבלת מספר – החזרת מספר החדר.
  2. האם החדר מאוכלס – החזרת ערך בוליאני.
3. פעולות set :
  1. איכלוס החדר – הרמת הדגל.
  2. פינוי החדר – הורדת הדגל.
4. הדפסה – הדפסת כל משתני המצב של החדר בפורמט הבא לדוגמא:

Number : 15

Size : 60 square meters

Is Occupied : true

### מחלקת כיתה

יש לממש מחלקה בשם class Class . מחלקה זו תמומש בקבצים Class.H, Class.C .

#### משתני מצב של כיתה:

1. כל משתני המצב של החדר ובנוסף:
2. יחס ילדים לגננת מקסימלי – משתנה ממשי (לאו דווקא שלם) שמהווה חסם עליון ליחס – number\_of\_children/number\_of\_teachers . לא ניתן לעבור יחס זה!
3. מספר מקסימלי של ילדים – משתנה שלם חיובי . בנוסף למגבלה על היחס , ישנה גם מגבלה על כמות הילדים בכיתה.
4. גיל הילדים – משתנה שלם חיובי שמגדיר את שכבר הגיל של הילדים בכיתה. למשל, אם שכבת הגיל היא 1-2 אז ערך המשתנה יהיה 1 .
5. רשימה של כל הילדים בכיתה - מומלץ להשתמש במחלקה vector אשר מוגדרת בספרייה <vector>
6. רשימה של כל הגננות בכיתה - גם כאן מומלץ להשתמש במחלקה vector .

#### הפעולות שנרצה לבצע על כיתה:

1. יצירת כיתה – נדרש לקבל מספר, גודל, יחס מקסימלי, גיל הילדים ומספר מקסימלי של ילדים.
2. פעולות get :
  1. קבלת מספר הגננות – החזרת מספר הגננות שיש בכיתה.
  2. קבלת שכבת הגיל.
  3. קבלת היחס הנוכחי – החזרת number\_of\_children/number\_of\_teachers . אם אין גננות אז גם אין ילדים ואז יוחזר יחס 0.
  4. קבלת טלפון של ילד – חיפוש ילד ברשימה לפי שם. אם הילד נמצא ברשימה המתודה תחזיר את מספר הטלפון של הוריו (מחרוזת). אחרת, המתודה תחזיר: "No Child".

**הטכניון - מכון טכנולוגי לישראל**  
הפקולטה להנדסת חשמל  
מבוא למערכות תוכנה

3. פעולות set:
1. הוספת גננת – הוספה לרשימה.
  2. הוספת ילד – בדיקה האם ניתן להוסיף ילד.
    1. היחס המקסימלי לא יופר.
    2. המספר המקסימלי של ילדים לא יופר.
- אם ניתן, אז יש להוסיף ילד לרשימה ולהחזיר הצלחה. אם לא, יש להחזיר כשלון (Result)
3. הורדת גננת – בדומה להוספת ילד, גם כאן יש לבדוק שהיחס המקסימלי לא יופר. אם לא אז יש להוריד את הגננת (לפי שם) מהרשימה. יש להחזיר הצלחה או כשלון.
  4. הורדת ילד – הורדת ילד מהרשימה לפי שם. יש להחזיר הצלחה או כשלון.
  5. קביעת ילד כחולה – חיפוש הילד בכיתה. אם נמצא נסמן אותו כחולה. יש להחזיר הצלחה או כשלון.
4. הדפסה – הדפסת כל משתני המצב של הכיתה בפורמט הבא לדוגמא:

*Printing class status :*

=====

*Number : 2*

*Size : 60 square meters*

*Is Occupied : true*

*Max number of children : 20*

*Number of children : 15*

*Number of teachers : 5*

*Max value for ratio : 3.25*

*Current ratio : 3*

*Children age range : 1 – 2*

*<EMPTY\_LINE>*

*Printing childrens status :*

=====

*<Print here the list of children>*

*<EMPTY\_LINE>*

*Printing teachers status :*

=====

*<Print here the list of teachers>*

*<EMPTY\_LINE>*

הערות:

1. שימו לב שבשורה השנייה, ה-13 וה-16 יש בדיוק 24 תווים של "שווה" (=).
2. ישנה שורה ריקה לפני הדפסת הילדים. **יש להדפיס אותה גם אם אין כיתות.**
3. אין להדפיס את רשימת הילדים (**שורות 13-16**) אם אין ילדים בכיתה.
4. הילדים יודפסו בסדר שבו הם נכנסו לכיתה.
5. אין להדפיס את רשימת הגננות (**שורות 17-20**) אם אין גננות בכיתה.
6. הגננות יודפסו בסדר שבו הן נכנסו לכיתה.

### מחלקת משרד

יש לממש מחלקה בשם class Office. מחלקה זו תמומש בקבצים Office.H, Office.C.

### משתני מצב של משרד:

1. כל משתני המצב של החדר ובנוסף:
2. המשרד עסוק – משתנה בוליאני שמחזיר אמת אם ורק אם המשרד עסוק. המשרד עסוק מהרגע שבו התקבלה הפקודה שיש ילד חולה עד שהילד משוחרר לביתו.

## הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל

מבוא למערכות תוכנה

3. שם הילד החולה – אם יש ילד חולה שעוד לא שוחרר לביתו, שמו יופיע במשתנה הזה

(מחרוזת). אם אין, המידע שם לא רלוונטי ואפשר לשים בו מה שרוצים.

**שימו לב:** אם המשרד עסוק, הוא לא יבצע אף פעולה עד שהילד החולה משתחרר (חוץ מפעולת שחרור הילד, כמובן). מכאן שלא ייתכן 2 ילדים חולים (או יותר).

4. רשימה של כל הכיתות בגן- מומלץ להשתמש במחלקה vector אשר מוגדרת בספרייה

<vector>

### הפעולות שנרצות לבצע על משרד:

- יצירת משרד – אין צורך בפרמטרים. משרד הוא תמיד חדר מספר 1, בגודל של 20 מ"ר. בנוסף, עם יצירתו, המשרד מאוכלס מיידית (פוקנצית איכלוס של החדר).
- הוספת כיתה – יצירת כיתה חדשה לפי פרמטרים נתונים. המספר הסידורי של הכיתה יהיה לפי החוק הבא:  $Class\_num = (number\_of\_classes) + 2$  וזאת מכיוון שלמשרד יש מספר סידורי של 1.
- שימו לב:** אין מניעה שלכיתות שונות יהיה את אותו המספר (זה יכול לקרות). לא ניתן להוסיף כיתה לשכבת גיל שכבר קיימת. אם יש ניסיון לעשות זאת יש להחזיר כשלון.
- הוספת ילד – הוספת ילד לכיתה לפי גיל הילד. ניתן להניח שגיל הילד תמיד מספר שלם חיובי.
- הוספת גננת – הוספת גננת לאחת הכיתות לפי האלגוריתם הבא:
  - אם יש מספר כיתות ריקות (אין ילדים ואין גננות) נוסיף גננת לאחת מהן (לפי סדר כניסת הכיתה לגן)
  - אחרת כל הכיתות מאוכלסות, ולכן נקצה את הגננת לכיתה לפי היחס  $[number\_of\_children/number\_of\_teachers]$  בצורה הבאה:
    - אם **כל הכיתות** בעלות יחס 0 (זה אומר שאין ילדים ויש גננות כי הכיתות מאוכלסות) – נקצה גננת לכיתה עם מספר הגננות הנמוך ביותר.
    - במקרה שיש יותר מכיתה אחת כזו - נקצה לפי סדר הכניסה של הכיתה לגן
    - אחרת, נקצה את הגננת לכיתה עם היחס הגבוה (הגרוע) ביותר.
    - במקרה שיש יותר מכיתה אחת כזו - נקצה לפי סדר הכניסה של הכיתה לגן.

לדוגמא:

הגן כרגע בעל 3 כיתות ריקות לגמרי. נסמן אותן במספרים 2,3,4 (בהתאמה לסדר הכניסה לגן).

הוספת גננת – כל הכיתות ריקות אז נוסיף לגן מספר 2.

הוספת עוד גננת – כיתה 3, 4 לא מאוכלסות – נוסיפה לכיתה 3.

הוספת עוד גננת – רק כיתה 4 לא מאוכלסת – נוסיפה לכיתה 4.

הוספת ילד לגן 2 – נניח שאפשר, אז מוסיפים

הוספת ילד לגן 3 – נניח שאפשר, אז מוסיפים

כעת המצב הוא כזה:

כיתה	2	3	4
מספר ילדים	1	1	0
מספר גננות	1	1	1

הוספת גננת – כל הכיתות מאוכלסות – נסתכל על היחס. לכיתות 2,3 יש יחס גדול משל 4.

נקצה לאחת מהן – נוסיפה לכיתה 2.

הורדת ילד - נניח הוא נמצא בגן 2. מורידים.

הורדת ילד - נניח הוא נמצא בגן 3. מורידים.

כעת המצב הוא כזה:

כיתה	2	3	4
מספר ילדים	0	0	0
מספר גננות	2	1	1

הוספת גננת - כל הכיתות מאוכלסות – נסתכל על היחס. כל הכיתות בעלות יחס 0. נוסיף

לפי מספר הגננות הקטן ביותר. לכיתות 3,4 יש רק גננת אחת. נוסיף גננת לכיתה 3.

## הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל

מבוא למערכות תוכנה

5. הורדת ילד – חיפוש ילד בכיתות לפי שם ושחרורו הבייתה (מחיקה מהרשימות).
  6. הורדת גננת - חיפוש גננת בכיתות לפי שם ושחרורה הבייתה (מחיקה מהרשימות).
  7. הורדת כיתה – חיפוש כיתה לפי שכבת גיל . אם נמצאה , נסיר אותה מהרשימה ונמחק אותה מהתכנית (delete). יש להחזיר הצלחה או כשלון.
  8. דיווח על ילד חולה – לפי האלגוריתם הבא:
    1. חיפש ילד לפי שם
    2. בדיקות תקינות של מספר הטלפון של הוריו לפי החוק :
      1. ישנם בדיוק 10 ספרות
      2. 2 הספרות הראשונות הן "05"
- שימו לב:** ניתן להניח שהמחרוזת שמכילה את הטלפון היא של ספרות , אין צורך לבדוק זאת. מומלץ להשתמש בפונקציות של מחלקת string.
3. אם הילד נמצא , והמספר תקין אז המשרד יהיה בסטאטוס עסוק ותודפס ההודעה הבאה (לדוגמא):

*Reporting sick child :*

*Name : Dani Din*

*Parent's phone number : 0541234567*

**שימו לב:** ברגע שמשרד נכנס לסטאטוס עסוק , כל בקשה ממנו תיענה בכישלון פרט להדפסה ולשחרור הילד שחולה.

9. הדפסה – הדפסת כל משתני המצב של הכיתה בפורמט הבא לדוגמא:

*Printing office status :*

=====

*Number : 1*

*Size : 20 square meters*

*Is Occupied : true*

*Office is busy : true*

*Sick child : Dani Din*

*<EMPTY\_LINE>*

*Printing class status :*

=====

.

.

.

*Printing class status :*

=====

.

.

.

הערות:

1. שימו לב שבשורה השנייה יש בדיוק 24 תווים של "שווה" (=)
2. אין להדפיס את הילד החולה (שורה 6) אם אין ילד חולה.
3. ישנה שורה ריקה לפני הדפסות הכיתות. יש להדפיס אותה גם אם אין כיתות.
4. אין להדפיס את רשימת הכיתות (שורות 8-9) אם אין כיתות בגן.
5. הכיתות יודפסו בסדר שבו הן נכנסו לגן.

**הטכניון - מכון טכנולוגי לישראל**  
הפקולטה להנדסת חשמל  
מבוא למערכות תוכנה



## חלק 4 - קלט ופלט

לרשותכם נתון הקובץ main.C חלקי, המבצע parsing של פקודות טקסט, ומתרגם אותן לפעולות שנדרשתם לממש על הגן. עליכם להשלים את החלקים החסרים בקובץ כך שתרגום הפקודות המוגדרות מטה יתבצע בצורה תקינה. שימו לב כי נתונות לשימושכם הפונק' הבאות:

tokenize - הפונק' מקבלת מחרוזת ורשימת מפרידים, ומחזירה vector של מחרוזות, כך שכל איבר בו הוא מחרוזת המכילה מילה, כאשר המפרידים בין המילים הם המפרידים המופיעים ברשימה (בדומה ל-strtok בשפת C).

stringToInt - פונק' המקבלת מחרוזת וממירה אותה לערך שלם (בדומה ל-atoi בשפת C).

stringToDouble - פונק' המקבלת 2 מחרוזות וממירה אותן למספר, לא בהכרח שלם. מחרוזת אחת מומרת לחלק השלם והשניה לחלק העשרוני.

**שימו לב:** הניחו שהמחרוזת שמתאימה לחלק העשרוני מכילה בדיוק 3 ספרות. למשל,

$10.25 = \text{stringToDouble}("10", "250")$

$5.0 = \text{stringToDouble}("5", "000")$

הפקודות שאתם נדרשים לתמוך בהן:

*addClass <size> <ratio> <max\_num\_children> <children\_age>*

השורה תגרום להוספת כיתה חדשה בעל הפרמטרים הנ"ל לגן. ניתן להניח שהמחרוזת **ratio** מייצגת מספר עשרוני כך שהחלק השברי בה הוא בעל 3 ספרות בדיוק, למשל – 3.250. במידה והקלט לא תקין (חסרים/עודף פרמטרים) או שהפקודה נכשלת - יש להדפיס הודעת שגיאה לערוץ השגיאות הסטנדרטי, ולא להוסיף את הכיתה לגן.

*removeClass <age>*

השורה תסיר את הכיתה ששכבת הגיל שלה הוא הפרמטר age. אם אין כיתה כזאת, יש להדפיס הודעת שגיאה לערוץ השגיאות הסטנדרטי.

*addChild <age> <name> <phone\_num>*

השורה תגרום להוספת ילד בעל הפרמטרים הנ"ל לגן. במידה והקלט לא תקין (חסרים/עודף פרמטרים) או שהפקודה נכשלת - יש להדפיס הודעת שגיאה לערוץ השגיאות הסטנדרטי, ולא להוסיף את הילד.

*addTeacher <age> <name> <seniority>*

השורה תגרום להוספת גננת בעלת הפרמטרים הנ"ל לגן. במידה והקלט לא תקין (חסרים/עודף פרמטרים) או שהפקודה נכשלת - יש להדפיס הודעת שגיאה לערוץ השגיאות הסטנדרטי, ולא להוסיף את הגננת.

*removeChild <name>*

השורה תגרום להוצאת ילד בעל השם הנ"ל מהגן. ניתן להניח, שאין 2 ילדים בעלי אותו שם בגן. במידה והקלט לא תקין (חסרים/עודף פרמטרים) או שהפקודה נכשלת - יש להדפיס הודעת שגיאה לערוץ השגיאות הסטנדרטי, ולא להסיר את הילד.

*removeTeacher <name>*

השורה תגרום להוצאת גננת בעל השם הנ"ל מהגן. ניתן להניח, שאין 2 גננות בעלי אותו שם בגן. במידה והקלט לא תקין (חסרים/עודף פרמטרים) או שהפקודה נכשלת - יש להדפיס הודעת שגיאה לערוץ השגיאות הסטנדרטי, ולא להסיר את הגננת.

*PrintKindergarten*

השורה תגרום להדפסת פרטי הגן. במידה ויש פרמטרים עודפים - יש להדפיס הודעת שגיאה לערוץ השגיאות הסטנדרטי, ולא להדפיס פרטי הרשת החברתית.

*sickChild <name>*

השורה תגרום לילד בעל השם הנ"ל להיות חולה. הגן ינסה להתקשר להוריו ויחכה עד שיוציאו אותו (ראה למעלה). במידה והקלט לא תקין (חסרים/עודף פרמטרים) או שהפקודה נכשלת - יש להדפיס הודעת שגיאה לערוץ השגיאות הסטנדרטי, לא לסמן את הילד כחולה ולאפשר לגן לקבל עוד פקודות.

**שימו לב:** אם צריך להדפיס הודעת שגיאה לערוץ השגיאות, אז הפורמט שלה יהיה כזה:

Failed - <command line>

למשל, אם הפקודה **removeChild dani** נכשלה, יודפס **Failed - removeChild dani**

## חלק 5 - תרגילון Bash

כתבו סקריפט בשם `remove_comments`, בן 10 שורות לכל היותר (לא כולל השורה `bin/bash/#!/` ולא כולל שורות תיעוד) המקבל פרמטר אחד - שם של תיקייה (שאותה הסקריפט יוצר). הסקריפט יוצר תיקייה חדשה, תחת התיקיה הנוכחית, עם הפרמטר כשם. לאחר מכן, הסקריפט מסיר הערות מכל הקבצים בתיקיה הנוכחית ומעתיק אותם לתיקיה החדשה שהוא יצר.

**שימו לב:** ההערות שיש להוריד הן שורות **שמתחילות** במחרוזת `"/` בלבד. כל שורה אחרת נשארת.

לדוגמה, נניח כי התיקיה הנוכחית מכילה 3 קבצים:

A.C A.H, main.C

הפקודה

`remove_comments new_dir`

תיצור תיקייה בשם `new_dir` ותעתיק אליו את 3 הקבצים ללא הערות מהפורמט הנ"ל.

**הערה:** קראו באינטרנט על `egrep` - ועל התווים המיוחדים `{*,$,^,-,+,?}` למשל, ניתן לקרוא על כך כאן:

<http://www.computerhope.com/unix/uegrep.htm>

**הערה 2:** הסקריפט הנ"ל שימושי כאשר מוסרים קטעי קוד נקיים ללקוח.

### 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. **שימו לב!** יש לקמפל את התוכנית בתוספת הדגל `-std=c++0x`, שורת הקומפילציה תיראה כך:

`g++ -g -Wall -std=c++0x my_program.c`

3. 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 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)
  - קובץ ה-`makefile` יוצר קובץ הרצה בשם הנדרש
4. 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.
5. שאלות בנוגע לתרגיל יש להפנות לפורום התרגיל ב-moodle בלבד.

**הטכניון - מכון טכנולוגי לישראל**  
 הפקולטה להנדסת חשמל  
 מבוא למערכות תוכנה

6. סיכום מפרט התרגיל:

סעיף	תיאור
נושא התרגיל	C++ בסיסי
תאריך ההגשה	יום ד', 13/6/2018 בשעה 23:59
המתרגל האחראי על התרגיל	יוחאי ריעאני syohai@campus
תיקייה המכילה קבצים לשימוש הסטודנטים	~eesoft/hmw/hmw4
קבצי הקוד הנתונים	defs.H main.C Person.H Person.C Child.H Child.C Teacher.H Teacher.C Room.H Room.C Class.H Class.C Office.H Office.C
קבצי הקלט והפלט הנתונים	error1.txt      input1.txt      output1.txt
הקבצים שיש להגיש	readme makefile remove_comments defs.H main.C Person.H Person.C Child.H Child.C Teacher.H Teacher.C Room.H Room.C Class.H Class.C Office.H Office.C
שם תכנית ההרצה הדרושה (הנוצרת ע"י makefile)	Kindergarten
דגשים מיוחדים	מומלץ להיעזר במחלקה vector. בפרט, קראו על המתודות erase, push_back, begin, end שלה. כמו כן, שימו לב כי מדובר במחלקת template - קראו מה נדרש מהטיפוס ששמור ב-vector. בנוסף, מומלץ להיעזר במחלקה string. קימפול בתוספת הדגל -std=c++0x

**בהצלחה!**

**הטכניון - מכון טכנולוגי לישראל**  
הפקולטה להנדסת חשמל  
מבוא למערכות תוכנה