

האוניברסיטה העברית בירושלים

בית הספר להנדסה ולמדעי המחשב ע"ש רחל וסלים בנין

סדנאות תכנות בשפת C ו-C++ (67312 ו-67317) C – תרגיל 1

תאריך הגשה יום רביעי, 13 לנובמבר, 2019, בשעה 23:55;
הגשה מאוחרת (בהפחתת 10 נקודות): יום חמישי, 14 לנובמבר, 2019, בשעה 23:55.

נושאי התרגיל: היכרות עם השפה, קומפילציה, משפטי pre-processor, משתנים, אריתמטיקה פשוטה, קלט & פלט, תנאים, לולאות, פונקציות ומערכים סטטים (ללא הקצאה דינמית).

1 רקע

האוניברסיטה העברית מעוניינת לנהל את הסטודנטים הרשומים בבית הספר להנדסה ולמדעי המחשב באמצעות מערכת ממוחשבת. האוניברסיטה זקוקה למימוש הפעולות הבאות:

- מציאת הסטודנט הטוב ביותר מבין רשימת סטודנטים;
 - מיון רשימת סטודנטים לפי פרמטרים שונים.
- האוניברסיטה פונה לעזרתכם, כדי שתחברו עבורה תוכנה שמבצעת את שתי הפעולות. בתרגיל זה תרכשו את מיומנויות הבסיס של כתיבת קוד בשפת C.
- במהלך התרגיל תכתבו את התוכנה `manageStudents`, שפותרת את האתגרים שהאוניברסיטה הציבה בפניכם. התרגיל מחולק לשתי מטלות, ומומלץ לשים לב לקשר בין שתיהן.

2 מטלה 1 – best student

הפעולה הראשונה ש-`manageStudents` נדרשת לבצע, היא מציאת הסטודנט הטוב ביותר מבין רשימה של סטודנטים שהמשתמש מזין ידנית.

- כדי לבצע פעולה זו, המשתמש יפעיל את התוכנה מה-CLI (Command Line Interface) עם הארגומנט "best", כך

```
$ ./manageStudents best
```

(כאשר התו "\$" מציין שורה שבה הוקלדה פקודה).

- התוכנה תבקש מהמשתמש להזין שורת קלט שמייצגת פרטים של סטודנט יחיד. הבקשה תודפס ל-stdout כך:

```
Enter student info. To exit press q, then enter
```

- המשתמש יזין שורת קלט אחת (ל-stdin) בפורמט הבא:

```
<ID> <name> <grade> <age> <country> <city>
```

כאשר כל שדה מופרד באמצעות Tab (התו "\t").
לדוגמה:

```
3845354610 Rhett Vargas 98 29 United-Kingdom London
```

- התוכנה תבדוק האם הקלט תקין (פירוט בהמשך).
- אם הקלט לא תקין, תודפס ל-stdout (בתרגיל זה, ההודעה לא תודפס ל-stderr) הודעת ERROR אינפורמטיבית שמדווחת למשתמש מהי צורת הקלט הנכונה, ומבקשת להזין קלט שוב (פירוט בהמשך).
- כשהמשתמש רוצה לסיים להזין נתונים עליו להזין את המחרוזת "q", ואז ללחוץ על Enter.
- בשלב זה התוכנה תדפיס ל-stdout את הטקסט "best student info is: " (שימו לב לרווח אחרי הנקודתיים), ואחריו (באותה שורה) את פרטיו של הסטודנט הטוב ביותר. הסטודנט הטוב ביותר מוגדר להיות זה שהגיע להשגים מרשימים בגיל צעיר, כלומר זה שעבורו הערך $\frac{grade}{age}$ גבוה ביותר.
- להלן דוגמה להרצה אפשרית של התוכנית:

```
$ ./manageStudents best
Enter student info. To exit press q, then enter
3888914775 Itzel Gardner 21 26 Iran Tehran
Enter student info. To exit press q, then enter
5496060426 Korbin Murillo 4 30 Pakistan Islamabad
Enter student info. To exit press q, then enter
3845354610 Rhett Vargas 98 29 United-Kingdom London
Enter student info. To exit press q, then enter
q
best student info is: 3845354610 Rhett Vargas 98 29 United-Kingdom
London
```

- (כאשר הסימן \$ מסמן את הפקודה שהוקלדה ב-terminal וטקסט הצבוע בצבע ירוק מסמן קלט מהמשתמש שלא יודפס על ידי התוכנה).

2.1 תקינות הקלט

השדות השונים בקלט צריכים לקיים את התנאים הבאים

- **ID:** מספר בין 10 ספרות שאינו מתחיל ב-0.
- **Name:** עשוי לכלול מילה אחת או יותר. מכיל אך ורק אותיות באנגלית (גדולות או קטנות), רווחים, או תו “-”. למרות שהשם יכול להכיל רווחים, **ניתן להניח** שהוא לא מתחיל ברווח ולא מסתיים ברווח.
- **Grade:** מספר שלם בין 0 ל-100 (כולל).
- **Age:** מספר שלם בין 18 ל-120 (כולל).
- **Country:** מכילה אך ורק אותיות באנגלית (גדולות או קטנות), או תו “-”.
- **City:** מכילה אך ורק אותיות באנגלית (גדולות או קטנות), או תו “-”.

אם אחד השדות או יותר אינו תקין, אין לקבל את פרטי הסטודנט. יש להדפיס ל-stdout (ולא ל-stderr) הודעה שמתחילה בתווים “ERROR:” (ERROR: בצירוף רווח יחיד) ולאחר מכן מפרטת את התנאים לקלט תקין. לאחר מכן, בשורה חדשה, יש להדפיס את מספר השורה שהזין המשתמש, שאינה תקינה. לדוגמה, אם המשתמש הזין 10 שורות קלט תקינות, ולאחר מכן ניסה פעמיים ברצף להזין שורות שבהן השדה city לא תקין, ההדפסה תראה כך:

```
Enter student info. To exit press q, then enter
3845354610 Rhett Vargas 98 29 United-Kingdom bad#city
ERROR: city can only contain alphabetic characters or '-'
in line 10
Enter student info. To exit press q, then enter
5496060426 Korbin Murillo 4 30 Pakistan bad#city
ERROR: city can only contain alphabetic characters or '-'
in line 11
```

שימו לב שהשורה הראשונה שהמשתמש מזין, מספרה 0. ניתן להניח שכל שורה שהמשתמש מזין תהיה באורך של עד 150 תווים (כולל תו השורה החדשה, “\n”, שמופיע בסוף השורה), ושכל שדה בשורה הוא באורך עד 40 תווים (לא כולל תווי Tab “\t”) (שמפרידים בין השדות השונים). אין צורך לוודא פרטים אלו, ולא תיבדקו עליהם.

2.2 דגשים למטלה 1

- השדות השונים בכל שורה מופרדים זה מזה ב-Tab “\t”. גם בסופו של השדה האחרון מופיע Tab. כאשר אתם מדפיסים שורות, הקפידו להדפיס אותן באותו אופן, כלומר כשהשדות מופרדים ב-Tab ולא בצורה אחרת (לדוגמה, ברווח), וגם אחרי השדה האחרון מופיע Tab. שימו לב: שדה לא נגמר עד שמופיע ה-Tab בסיומו. אם מופיעים תווים לא תקינים לפני ה-Tab, הערך בשדה אינו תקין.
- יש לסיים כל הדפסה של שורה כלשהי בתו newline “\n”.

- אם יותר מסטודנט אחד מקבל ערך $\frac{grade}{age}$ מקסימלי, יש להדפיס את הסטודנט הראשון מבניהם שהוקלד על ידי המשתמש.
- **ניתן להניח** שהמשתמש יזין לכל היותר 5000 שורות קלט. אין צורך לוודא פרט זה, ולא תיבדקו עליו.

3 מטלה 2 – merge sort and quick sort

שתי הפעולות הנוספות שהתוכנה manageStudents מבצעת הן מיון של רשימת סטודנטים שהמשתמש מזין, בשיטות שונות.

- כדי למיין את רשימת הסטודנטים **לפי הציונים שקיבלו** (בסדר עולה), באמצעות אלגוריתם merge sort, המשתמש יפעיל את התוכנה מה־cli עם הארגומנט “merge”, כך:

```
$ ./manageStudents merge
```

- כדי למיין את רשימת הסטודנטים **לפי סדר אלפביתי של שמם** (בסדר עולה), באמצעות אלגוריתם merge sort, המשתמש יפעיל את התוכנה מה־cli עם הארגומנט “quick”, כך:

```
$ ./manageStudents quick
```

- עם הפעלת התוכנה, המשתמש יתבקש להזין את פרטי הסטודנטים (ל־stdin) באופן זהה לזה שבמטלה 1. יש לטפל בשורות הקלט של המשתמש באותו אופן בדיוק כמו במטלה 1, כולל בדיקת תקינות הקלט.
- אם אחד השדות או יותר אינו תקין, אין לקבל את פרטי הסטודנט ואין להכניס אותו לרשימה הממויינת. יש להדפיס ל־stdout הודעת שגיאה כמו במטלה 1.
- ניתן להניח שהמשתמש מזין לכל היותר 5000 שורות קלט. אין צורך לוודא פרט זה, ולא תיבדקו עליו.
- לאחר שהמשתמש יסיים להזין שורות קלט (ויסמן זאת בהזנת המחרוזת “q”, כמו במטלה 1), התוכנה תתחיל בתהליך המיון.
- על התוכנה למיין את רשימת הסטודנטים שהמשתמש הקליד, ולהדפיס ל־stdout את הרשימה הממויינת.
- **שימו לב:** חובה לממש את המיונים באמצעות אלגוריתמים merge sort ו־quick sort בהתאמה. תרגיל שיעשה שימוש בסוגי מיונים שונים יפסל וינוקד בציון 0. על כך לא תהא זכות ערעור.

3.1 דגשים למטלה 2

- יש להדפיס את הרשימה הממויינת כשהשורות מופיעות כפי שהן מופיעות בקלט, כלומר בפורמט:

```
<ID> <name> <grade> <age> <country> <city>
```

(כשהשדות השונים מופרדים זה מזה ב־TAB (\t), ובסוף כל שורה מופיע תו new line ((\n)).

4 דגשים כללים לתרגיל

- בכל מקרה שבו התוכנה מסיימת לפעול בהצלחה, יש להחזיר מהפונקציה main את הקוד 0. אם התוכנה נכשלת ונאלצת לעצור מסיבה כלשהי בלי שהשלימה את משימתה, יש להחזיר מהפונקציה main קוד שגיאה (ערך יציאה ששונה מ-0).
- במקרה שהתוכנה מופעלת עם ארגומנט שאינו מתאים לאף אחת משלוש הפעולות שמתוארות בתרגיל, או שניתן לתוכנה יותר מארגומנט אחד, יש להדפיס ל-stdout (ולא ל-stderr) הודעה שמתחילה ב-"USAGE: " (כלומר המילה USAGE, בצירוף נקודתיים ולאחריה רווח יחיד). לאחר מכן יש לכתוב הסבר על דרך ההפעלה הנכונה של התוכנה (הנוסח נתון לשיקולכם). לאחר מכן יש לצאת מהתוכנה עם קוד 1.
- מומלץ לקרוא לפני תחילת התרגיל על הזרם stdin שממנו קוראים קלט, ועל הזרם stdout שאליו מדפיסים פלט. אין צורך להבין מונחים אלה לעומק, אבל צריך לדעת כיצד להשתמש בהם.
- כל ההדפסות בתרגיל, כולל הודעות שגיאה, יודפסו אך ורק ל-stdout. אין להדפיס שום תוכן ל-stderr.
- כל הודעות השגיאה חייבות להתחיל ב-"ERROR: ". נוסח ההודעה לאחר מכן נתון לשיקולכם, אך מצופה שיהיה אינפורמטיבי ויאפשר למשתמש לטפל בבעיה.
- בתרגיל זה מומלץ להשתמש בפונקציות scanf, sscanf, gets. אם משתמשים בפונקציות אלו יש לוודא שהן מסיימות את פעולתן בהצלחה. **אין להשתמש בפונקציות שאינן בטוחות, כדוגמת scanf (תוכלו לקרוא איסור זה בנהלים להגשת תרגילים).**

5 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!
- **שימו לב:** חובה לממש את מטלה 2 באמצעות אלגוריתמים merge sort ו-quick sort. תרגיל שיעשה שימוש בסוגי מיונים שונים יפסל וינוקד בציון 0. על כך לא תהא זכות ערעור.
- כתבו את כל ההודעות שבהוראות התרגיל בעצמכם. העתקת ההודעות מהקובץ עלולה להוסיף תווים מיותרים ולפגוע בבדיקה האוטומטית, המנקדת את עבודתכם.
- בשפת C יש פונקציות רבות שמיועדות לעבודה עם קלט. אין צורך להמציא מחדש את הגלגל! לפני תחילת העבודה על התרגיל, מומלץ לחפש באינטרנט את הפונקציות המתאימות ביותר לקבלת קלט מהמשתמש, להדפסת קלט, עיבוד קלט מסוגים שונים וכו'. ודאו שכל הפונקציות שבהן אתם משתמשים מתאימות לתקינה C99, וכי אתם יודעים כיצד הן מתנהגות בכל סיטואציה.
- כאמור, למעט הודעת ה-Usage, כל הודעות השגיאה חייבות להתחיל במחרוזת "ERROR: ". נוסח ההודעה לאחר מכן נתון לשיקולכם, אך מצופה שיהיה אינפורמטיבי ויאפשר למשתמש לטפל בבעיה. הודעות השגיאה צריכות להיות בנות שורה אחת.

- פתרון בית הספר זמין בנתיב:

```
~labcc/www/c_ex1/manageStudents
```

- עליכם ליצור קובץ tar הכולל **אך ורק** את הקובץ `manageStudents.c`. ניתן ליצור tar כדרוש על ידי:

```
$ tar -cvf c_ex1.tar manageStudents.c
```

- כדי להדר את התרגיל מהקובץ `manageStudents.c` לקובץ בינארי בשם `manageStudents`, תוכלו להשתמש בפקודה הבאה:

```
gcc -Wextra -Wall -Wvla -std=c99 -lm manageStudents.c -o  
manageStudents
```

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

```
$ ~labcc/www/codingStyleCheck <code file or directory>
```

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

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

```
~labcc/www/c_ex1/presubmission
```

בהצלחה!!