

האוניברסיטה העברית בירושלים
בית הספר להנדסה ולמדעי המחשב ע"ש רחל וסלים בנין

סדנת תכנות בשפת C ו-C++ (67315) C - תרגיל 2

תאריך הגשה: 21 לאפריל, 2021, בשעה 23:55;

נושאי התרגיל: מערכים, מצביעים, structs, הקצאה דינאמית, ניהול זיכרון, אריתמטיקה של פוינטרים ומיון.

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 עבודה תוכנה שמבצעת את שתי הפעולות.

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 את מספר התלמידים בבית הספר. הבקשה תודפס ל-`stdout` כך:

```
Enter number of students. Then enter
```

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

<Number of students>

10 - לדוגמה:

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

Enter student info. Then enter

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

<ID>, <grade>, <age>

כאשר כל שדה מופרד על ידי פסיק יחיד (התו ",").

3845354610,98,29 - לדוגמה:

- 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). הקלט שאינו תקין לא נשמר.
- הזנת הנתונים מסתיימת לאחר שהוזנו n שורות קלט תקינות, כאשר n הוא מספר התלמידים שהוזן בשורת הקלט הראשונה.
- 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 $\frac{grade}{age}$ (float) גבוה ביותר.
- להלן דוגמה להרצה אפשרית של התוכנית:

```
$ ./manageStudents best
Enter number of students. Then enter
3
Enter student info. Then enter
3888914775,21,26
Enter student info. Then enter
5496060426,4,30
Enter student info. Then enter
3845354610,98,29
best student info is: 3845354610,98,29
```

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

2.1 תקינות הקלט

השדות השונים בקלט צריכים לקיים את התנאים הבאים:

- Number of students: מספר שלם גדול מ-0.
- ID: מספר בן 10 ספרות בדיוק, שאינו מתחיל ב-0.
- Grade: מספר שלם בטווח [0,100].
- Age: מספר שלם בטווח [18,120].

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. ההדפסה תראה כך:

```
Enter number of students. Then enter
1
Enter student info. Then enter
3845354610,98,-2
ERROR: Age should be an integer between 18 and 120 (includes).
Enter student info. Then enter
0496060426,4,30
ERROR: Id must have 10 digits and can not start with 0.
Enter student info. Then enter
3845354610,98,29
best student info is: 3845354610,98,29
```

2.1.1 הנחות מקדימות

את ההנחות הבאות ניתן להניח לגבי הקלט:

- כל שורה שהמשתמש מזין היא באורך של עד 60 תווים (כולל תו שורה חדשה, `\n`, שמופיע בסוף השורה).
- כל שדה בקלט (ת,ז, ציון וגיל) הוא באורך של עד 20 תווים (כולל תו שורה חדשה, `\n`, שמופיע בסוף השורה).
- שורת הקלט של מספר הסטודנטים בבית הספר מכילה מספר שלם בלבד, כך גם עבור השדות גיל וציון.
- שורת קלט של פרטי סטודנט מכילה בדיוק 3 שדות, כך שבין כל שדה יש פסיק בודד (,).
- שורת הקלט לא מכילה כלל תווי 'רווח' (space).
- כשאתם כותבים את התרגיל, אין צורך להתחשב בשורות שאינן מתאימות להנחות אלו.

2.1.2 דגשים למטלה 1

• עליכם להגדיר struct בשם *Student* שמכיל את השדות *long int id, int grade, int age*.

• עליכם להקצות מערך דינאמי של structs מסוג *Student*. על המערך להיות באורך מספר הסטודנטים.

• במידה ואירעה שגיאה בעת ההקצאה, עליכם לצאת מהתוכנית עם קוד 1.

• המידע שהתקבל כקלט עבור כל סטודנט ישמר ב־struct במערך, על פי הסדר בו נקלט.

• עליכם ליצור פונקציה שמקבלת כקלט פוינטר לתחילת המערך ופוינטר לסוף המערך (מיקום הפוינטרים בתחילת המערך ובסוף המערך מוצגים באיור 1), ובסיום מדפיסה את פרטיו של הסטודנט המצטיין כפי שפורט למעלה. הקריאה לפונקציה תבצע לאחר מילוי המערך (כלומר מספר השורות הקלט התקינות שהתקבלו שווה למספר התלמידים בבית הספר).

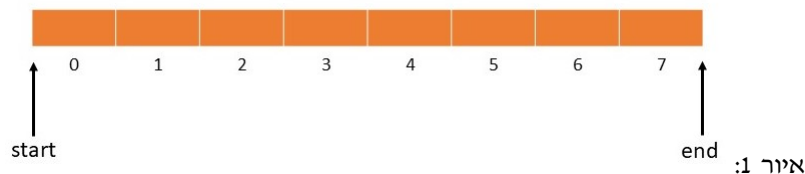
• החתימה של הפונקציה היא:

```
void best_student(Student *start, Student *end)
```

• **שימו לב:** אין להשתמש באופרטור סוגריים מרובעים ([]) בפונקציה *best_student*, מי שיעשה כן צפוי לאבד נקודות. עליכם להשתמש באריתמטיקה של פוינטרים במעבר על המערך.

• אם יותר מסטודנט אחד מקבל ערך $\frac{grade}{age}$ (float) מקסימלי, יש להדפיס את הסטודנט הראשון מבניהם שהוקלד על ידי המשתמש.

• בכל הדפסה של שורה כלשהי, יש לסיים בתו `\n` (newline).



3 מטלה 2 - bubble 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 *bubble sort*, המשתמש יפעיל את התוכנה מה־cli עם הארגומנט "bub-ble", כך:

```
$ ./manageStudents bubble
```

- כדי למיין את רשימת הסטודנטים **לפי גיל** (בסדר עולה), באמצעות אלגוריתם quick 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.
- לאחר שהמשתמש יסיים להזין שורות קלט (כלומר מספר השורות הקלט התקינות שהתקבלו שווה למספר התלמידים בבית הספר), התוכנה תתחיל בתהליך המיון כפי שמפורט בשלבים הבאים.
- על התוכנה למיין את רשימת הסטודנטים שהמשתמש הקליד, ולהדפיס ל־stdout את הרשימה הממויינת.
- עבור אלגוריתם bubble sort: עליכם ליצור פונקציה שמקבלת כקלט פוינטר לתחילת המערך ופוינטר לסוף המערך (ראו איור 1), מבצעת על המערך מיון בועות, ובסיום מדפיסה את הרשימה הממויינת. הקריאה לפונקציה תתבצע לאחר מילוי המערך.

- החתימה של הפונקציה היא:

```
void bubble_sort(Student *start, Student *end)
```

- עבור אלגוריתם quick sort: עליכם ליצור פונקציה שמקבלת כקלט פוינטר לתחילת המערך ופוינטר לסוף המערך (ראו איור 1), מבצעת על המערך מיון מהיר, ובסיום מדפיסה את הרשימה הממויינת. הקריאה לפונקציה תתבצע לאחר מילוי המערך. שימו לב, הפונקציה quick_sort **חייבת לקרוא** לפונקציה partition.

- החתימות של הפונקציות הן:

```
void quick_sort(Student *start, Student *end)
Student *partition(Student *start, Student *end)
```

- **שימו לב:** אין להשתמש באופרטור סוגריים מרובעים ([]) בפונקציות bubble_sort, quick_sort ו־partition, מי שיעשה כן צפוי לאבד נקודות. עליכם להשתמש באריתמטיקה של פוינטרים במעבר על המערך.
- **שימו לב:** חובה לממש את המיונים באמצעות אלגוריתמים bubble 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

- אם לשני סטודנטים יש ערך זהה בשדה המיון, אין חשיבות לסדר ההדפסה שלהם.
- יש להדפיס את הרשימה הממויינת כשהשורות מופיעות כפי שהן מופיעות בקלט, כלומר בפורמט:

<ID>, <grade>, <age>

new line
new line

- תוכלו לקבל אינטואיציה ל-Bubble Sort באמצעות ההדגמה הזמינה בקישור:
<https://www.youtube.com/watch?v=nmhjrI-aW5o>
- באופן זה, תוכלו לקבל אינטואיציה ל-Quick Sort באמצעות ההדגמה הזמינה בקישור:
<https://www.youtube.com/watch?v=PgBzjICcFvc>

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 1.
- חובה לשחרר את כל המשאבים שהוקצו במהלך הריצה לפני היציאה מהתוכנית. כל פעולות ההקצאה והשחרור צריכות להתבצע כפי שנלמד בכיתה.
- 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.
- למעט הודעת ה-Usage המוזכרות למעלה, כל הודעות השגיאה חייבות להתחיל ב-"ERROR:". נוסח ההודעה לאחר מכן נתון לשיקולכם, אך מצופה שיהיה אינפורמטיבי ויאפשר למשתמש לטפל בבעיה.
- כל הודעות ה-Error וה-Usage צריכות להיות בנות שורה אחת.
- בתרגיל זה בלבד, על מנת להקל עליכם, כל ההדפסות כולל הודעות השגיאה יודפסו ל-stdout (לדוגמה, ניתן לעשות זאת על ידי שימוש בפונקציה printf). לכן, בתרגיל זה אין להדפיס שום תוכן ל-stderr, ומי שידפיס תוכן ל-stderr צפוי לאבד נקודות.
- בתרגיל זה מומלץ להשתמש בפונקציות scanf, sscanf, fgets. 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:** אין להשתמש באופרטור סוגריים מרובעים ([]) בפונקציות quick_sort, bubble_sort, best_student, partition, מי שיעשה כן צפוי לאבד נקודות. עליכם להשתמש באריתמטיקה של פוינטרים במעבר על המערך.
 - **שימו לב:** חובה לממש את מטלה 2 באמצעות אלגוריתמים quick ו-bubble 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.
 - יש להגיש את הפתרון בגיטהאב-האוניברסיטאי לפי נהלי ההגשה שהועלו למודל.
 - יש להגיש אך ורק את הקובץ manageStudents.c.
 - כחלק מהבדיקה האוטומטית תיבדקו על סגנון כתיבת קוד.
 - כדי לקמפל את התוכנית מהקובץ manageStudents.c תוכלו להשתמש בפקודה הבאה:
- ```
gcc -Wall -Wextra -Wvla -std=c99 manageStudents.c -o manageStudents
```
- שימו לב - הבדיקות האוטומטיות רצות על גבי מחשבי בית הספר, לכן וודאו כי הפתרון שלכם רץ ועובד על גבי מחשבי בית הספר.
  - שימו לב - ודאו כי הפתרון שלכם עובר את הפריסאבמיט ללא שגיאות או אזהרות, כשלוך בקומפילציה או בפריסאבמיט יגרוור ציון 0 בתרגיל.

**בהצלחה!!!**