

קורס תכנות 2 – תרגיל בית 4

מועד ההגשה: 24/05/16 בשעה 23:55.

הנחיות כלליות:

קראו בעיון את השאלות והקפידו שהתכניות שלכם יפעלו בהתאם לנדרש.
זכרו לפעול לפי כללי ההגשה המפורסמים באתר.

הערה חשובה: שימו לב שההדפסות שלכם הן זהות בדיוק לדוגמאות ההרצה. כולל רווחים וכולל שורות חדשות. ירדו נקודות על פורמט לא תקין.

שימו לב: את התרגיל יש לפתור לבד!

חישוב ציונים בקורס

בתרגיל הזה אנו נבנה תוכנית שתקבל את ציוני כל העבודות של כל אחד מהסטודנטים בקורס בקובץ text. ובאמצעות התוכנית תיצור קובץ text חדש ובו יהיה הממוצע של כל סטודנט לצד תעודת הזהות שלו וכן מספר נתונים כללים על ציוני הכיתה.

לפני שנתחיל בכתיבת הקוד נראה את המבנה הדרוש לקבצי הקלט והפלט:

קובץ הקלט:

קובץ הקלט נתון וממנו יש לקרוא את 4 חלקים הבאים (האחרון עשוי לא להופיע בקובץ - אופציונאלי):

חלק 1 – מספר הסטודנטים בקורס:

בשורה הראשונה יינתן מספר הסטודנטים בקורס, התואם את מספר הסטודנטים המופיעים בהמשך הקובץ. השורה הראשונה תהיה לפי המבנה שמוצג להלן:

The number of students: X

כאשר X הוא מספר הסטודנטים בקורס.

חלק 2 – מספר הציונים לכל סטודנט בקורס ומשקלם:

בשורה השנייה יינתן מספר העבודות שניתנו בקורס כולל מבחן, כלומר מספר הציונים שיש לכל אחד מהסטודנטים המשפיעים על ציונו הסופי. השורה השנייה תהיה לפי המבנה שמוצג להלן:

The number of grades: Y

כאשר Y הוא מספר הציונים/העבודות כולל המבחן בקורס.

בשורה השלישית יינתנו משקלי הציונים של כל עבודה באחוזים כאשר רווח יחיד מפריד ביניהם. כמו כן ניתן להניח שמספר המשקלים שיינתנו בשורה השלישית הוא המספר Y שנקלט בשורה השנייה של קובץ הטקסט. לדוגמא אם $Y=3$, אז השורה השלישית תהיה במבנה:

A% B% C%

חלק 3 – ציוני הסטודנטים בקורס:

מהשורה הרביעית והלאה יוזנו מספרי תעודות הזהות של הסטודנטים וציוניהם בכל אחד מהעבודות שניתנו בקורס כולל המבחן, בסדר בו הוזנו משקלי הציונים, כלומר הציון שמשקלו A יוזן ראשון וכך הלאה. כך שבכל שורה יוזנו תעודת זהות של הסטודנט ו-Y ציוניו של הסטודנט הנ"ל, בצורה הבאה:

123456789 a b c

כאשר 123456789 היא תעודת הזהות של הסטודנט, a ציונו בעבודה הראשונה שמשקלה A אחוזים, וכן הלאה. כאן הדוגמא תקפה למקרה שנקלט $Y=3$. כמו כן, כל סטודנט יהיה בשורה חדשה.

ניתן להניח שהקובץ תקין, כלומר 3 השורות הראשונות מופיעות תמיד במבנה הנתון, כמו כן ניתן להניח שגם X וגם Y מספרים שלמים חיוביים גדולים ממש מ-0. ומהשורה הרביעית והלאה, אם נקלט שמספר הסטודנטים הוא X אז אכן יוזנו X שורות בדיוק במבנה שהוצג לעיל, שורה לכל סטודנט. וכן אם נקלט שמספר הציונים הוא Y אז בכל שורה יהיה בתחילה מספר הזהות של הסטודנט ולאחריו Y מספרים בדיוק המתארים את ציוני הסטודנט.

חלק 4 (אופציונאלי) – פקטור:

כמו כן ישנה שורה נוספת אופציונאלית (כלומר היא יכולה או להופיע או לא להופיע) בסוף הקובץ, כלומר לאחר שהוזנו כל X הסטודנטים וציוניהם. שורה זו מתארת את הפקטור ואופן הפעלתו על ציונם הסופי של הסטודנטים. מבנה שורה זו, במידה והיא כן מופיעה, הוא:

There is a factor of: ZO

כאשר Z הוא מספר שלם ומציין את גודל הפקטור ו-O הוא האופרטור הקובע את אופן הפעלת הפקטור, והם צמודים אחד לשני ללא רווח, כאשר Z מופיע קודם ולאחריו O. לאופרטור O שתי אפשרויות:

התו: '+' – במקרה זה יש להוסיף Z לממוצע של כל סטודנט.

התו: '%' – במקרה זה יש להוסיף לממוצע של כל סטודנט Z אחוזים מהממוצע שלו.

שימו לב, שבכל אחת מהשיטות, ממוצע ציוניו של הסטודנט לא יכול לעבור את הציון 100 לאחר הפעלת הפקטור. כמו כן, את הפקטור יש להפעיל על ממוצע ציוניו של הסטודנט לפני שהוא עוגל למספר שלם.

קובץ הפלט:

את קובץ הפלט יש ליצור ובו צריכים להיות 4 חלקים:

חלק 1 – ציונים סופיים בקורס:

לפני הדפסת הציונים הסופיים בקורס תודפס השורה הבאה בראש קובץ הפלט:

The final grades in the course:

לאחר מכן, קובץ הפלט יכיל את הציון הממוצע של כל סטודנט לצד תעודת הזהות שלו במבנה הבא:

123456789 final grade is A

כאשר 123456789 היא תעודת הזהות של הסטודנט ו-A הינו ציונו הסופי (מספר שלם) של הסטודנט, לאחר הפקטור, במידה וניתן (הפקטור ניתן לפני שמעגלים את המספר). את המספר A יש לעגל למספר השלם הקרוב ביותר, שארית 0.5 יש לעגל כלפי מעלה. זה יהיה מעתה הציון הסופי שיינתן לסטודנט (כלומר הציון המעוגל) וכל שאר החישובים המתוארים בהמשך מתייחסים לציון מעוגל זה (כפי שנהוג באוניברסיטה לעגל את הציון של כל תלמיד אחרי הפקטור ומעכשיו כל החישובים יהיו ביחס לציון זה כמו הממוצע וכו', ראו להלן). כמו כן, ציונו של כל סטודנט יהיה בשורה חדשה.

חלק 2 – ממוצע הציונים בקורס:

לאחר הדפסת הציונים הסופיים של כל הסטודנטים בקורס יש להדפיס את הממוצע בקורס (לאו דווקא מספר שלם – השתמשו ב-g% לפורמט ההדפסה). מבנה השורה הוא:

The average grade is A

כאשר A הוא ממוצע הציונים בקורס והוא צריך להיות בפורמט g%.

חלק 3 – נכשלים בקורס:

בנוסף לכך, יש לחשב את מספר הנכשלים בקורס ולציין את מספרי תעודות הזהות של הסטודנטים שנכשלו. נכשל בקורס מוגדר כציון סופי הנמוך ממש מ-60, כלומר עובר זה ציון סופי 60 ומעלה. מבנה השורה צריך להיות, במידה ואף סטודנט לא נכשל:

The number of failures is 0

ואז גם אין צורך להדפיס תעודות זהות ויש לעבור לחלק הבא. אולם, אם יש סטודנט אחד או יותר שנכשלו, מבנה השורה יהיה:

The number of failures is F. The students who have failed:

כאשר F הינו מספר הסטודנטים שנכשלו בקורס. שימו לב שנוספו עוד מילים לשורה ולפיכך גם '!' (נקודה) שתפריד בין שני חלקי המשפט. במצב זה בו יש סטודנט אחד או יותר שנכשלו יש צורך להדפיס עוד שורה בה מצוינים מספרי תעודות הזהות של הסטודנטים שנכשלו, מופרדים ב- ', ' ו-' לאחריו (פסיק ואז רווח). למשל אם $F=2$, אז השורה הנוספת אותה יש להדפיס, תהיה לפי המבנה:

123456789, 987654321

כאשר 123456789 ו-987654321 הם מספרי תעודות זהות של הסטודנטים שנכשלו (כאמור $F=2$).

חלק 4 – טווח ציונים בקורס:

לבסוף, עליכם להדפיס את הציון המקסימלי שהושג בקורס, במבנה הבא:

The highest grade is M

כאשר M הינו הציון המקסימלי שהושג בקורס (מספר שלם). ולאחר מכן, להדפיס בשורה חדשה את הציון המינימלי שהתקבל בקורס, במבנה הבא:

The lowest grade is N

כאשר N הינו הציון המינימלי שהתקבל בקורס (מספר שלם). יש לשים לב שאין שורה חדשה לאחר הדפסת הציון המינימלי שהתקבל בקורס, שכן זוהי ההדפסה האחרונה.

הארגומנטים של התוכנית:

אנו נקבל 3 פרמטרים ב-argv של פונקציית ה-main:

- הפרמטר הראשון (argv[1]): יהיה השם של קובץ הקלט ממנו אנו רוצים לקרוא.
- הפרמטר השני (argv[2]): יהיה השם של קובץ הפלט אותו יש ליצור ויש להדפיס אליו.
- הפרמטר השלישי (argv[3]): יהיה מספר הסטודנטים בקורס שמכניס המשתמש.

יש לציין שבנוסף לארגומנט השלישי האומר מהו מספר הסטודנטים בקורס, אנו גם קולטים את מספר הסטודנטים בקורס מהשורה הראשונה בקובץ הקלט. במידה ושניהם זהים אז יש להדפיס לקובץ הפלט את 4 החלקים שתיארנו לעיל. אולם, במידה ואין התאמה בין שניהם, כלומר הם לא זהים, אז במקום להדפיס את 4 החלקים של קובץ הפלט כפי שתיארנו לעיל יש להדפיס את השורה:

The number of students is not compatible with the argument

בלבד ובמבנה הזה. ללא ירידת שורה בסופה, שכן זוהי ההדפסה האחרונה.

כתיבה מודולרית:

לצורך בניית התרגיל, נבנה מודול (שיכיל קובץ c וקובץ h בשמות: Ex4_q1_123456789.c ו- Ex4_q1_123456789.h בהתאמה) שיכלול מספר פונקציות עזר לצורך עבודה עם מבנה Student שאותו נגדיר לצורך ביצוע התרגיל, וכן מספר פונקציות עזר ליצירת מערכי עזר וחישובים אחרים. את המודול הנ"ל Ex4_q1_123456789.h, נייבא למודול הראשי (שיכיל קובץ c וקובץ h בשמות: Ex4_q2_123456789.c ו- Ex4_q2_123456789.h בהתאמה) שיכיל את פונקציית ה-main ועוד מספר פונקציות לטיפול בקבצים.

הגדרת המבנה Student שתופיע בקובץ ה-header: Ex4_q1_123456789.h, מובאת להלן:

```
typedef struct student{
    int id;
    int* grades;
    double avg;
} Student;
```

כאשר:

- id – הינו מספר תעודת הזהות של הסטודנט.
- grades – הינו מערך שיכיל את ציוניו של הסטודנט.
- avg – ממוצע הציונים של הסטודנט כפי שיוגדר להלן:

חישוב ממוצע ציוניו של סטודנט (לפני ביצוע פקטורים למיניהם) יהיה בהתאם לחלק 2 בקובץ הקלט בו קיבלנו את מספר הציונים ומשקלם על ציוניו במערך grades. החישוב יתבצע בצורה הבאה:

- אם סכום המשקלים שנקלטו שווה ל-100% אז ממוצע ציוניו של הסטודנט יהיה ממוצע משוקלל רגיל. לדוגמא:

עבור משקלים: 20% 30% 50% וציונים 81 72 85. ממוצע הציונים יהיה:

$$avg = 85 * 0.5 + 72 * 0.3 + 81 * 0.2 = 80.3$$

- אם סכום המשקלים שנקלטו קטן מ-100% אז ממוצע ציוניו של הסטודנט יהיה ממוצע משוקלל על ציוניו של הסטודנט עם המשקלים המתאימים ובנוסף ציון 100 עם משקל שישלים את סכום המשקלים ל-100%. לדוגמא:

עבור משקלים: 20% 25% 50% וציונים 81 72 85. ממוצע הציונים יהיה:

$$avg = 85 * 0.5 + 72 * 0.25 + 81 * 0.2 + 100 * 0.05 = 81.7$$

- אם סכום המשקלים שנקלטו גדול מ-100% אז ממוצע ציוניו של הסטודנט יהיה ממוצע משוקלל על ציוניו של הטובים ביותר של הסטודנט עם המשקלים המתאימים. במידה ויש להקטין את המשקל של הציון שישלימו ל-100%, יש לעשות זאת. לדוגמא:
- עבור משקלים: 25% 30% 50% וציונים 81 72 85. ממוצע הציונים יהיה:
- $$avg = 85 * 0.5 + 72 * 0.25 + 81 * 0.25 = 80.75$$

כעת, לאחר שכל הדרישות הובהרו ניגש לכתיבת הפונקציות, לפי החלוקה למודולים הבאים:

ב- Ex4_q1_123456789 :

1. כתבו את הפונקציה בעלת החתימה הבאה:

```
Student initStudent (int id, int* grades, double avg)
```

הפונקציה מקבלת מספר id – משתנה מטיפוס int. grades - מצביע למשתנה מטיפוס int ו- avg - משתנה מטיפוס double. הפונקציה מחזירה משתנה מטיפוס Student ששדותיו אותחלו לפי הארגומנטים של הפונקציה.

2. כתבו את הפונקציה בעלת החתימה הבאה:

```
Student* initStudentArray (int numofStudents)
```

הפונקציה מקבלת מספר numofStudents – משתנה מטיפוס int. הפונקציה **מקצה דינאמית** מערך של סטודנטים בגודל numofStudents ומאתחלת כל איבר בו לסטודנט ששדותיו id=0, grades=NULL, avg=0.0. הפונקציה initStudent. הפונקציה מחזירה את המערך מטיפוס Student.

3. כתבו את הפונקציה בעלת החתימה הבאה:

```
double* createWeightsArray (char* str, int numofGrades)
```

הפונקציה מקבלת מחרוזת str שהיא השורה השלישית בקובץ הקלט ומספר numofGrades – משתנה מטיפוס int. הפונקציה **מקצה דינאמית** מערך של משקלים מטיפוס double בגודל numofGrades. הפונקציה שמה את משקלי הציונים בתוך מערך המשקלים בצורה של מספר עשרוני בין 0 ל-1. הפונקציה מחזירה את המערך מטיפוס double.

4. כתבו את הפונקציה בעלת החתימה הבאה:

```
double calcAvgGrade (double* weightsArr, int* grades, int numofGrades)
```

הפונקציה מקבלת מערך של משקלים weightsArr מטיפוס double, מערך של ציונים grades מטיפוס int, ואת אורכי המערכים - המספר numofGrades – משתנה מטיפוס int. הפונקציה מחזירה את ממוצע הציונים המשוקלל בהתאם לדרישות הקורס שהובהרו לעיל.

5. כתבו את הפונקציה בעלת החתימה הבאה:

```
Student getStudentFromString (char* str, double* weightsArr, int numofGrades)
```

הפונקציה מקבלת מחרוזת str שהיא שורה במבנה המתואר בחלק 3 בקובץ הקלט, מערך של משקלים weightsArr מטיפוס double, ואת אורכו - המספר numofGrades – משתנה מטיפוס int. הפונקציה **מקצה דינאמית** מערך של ציונים מטיפוס int בגודל numofGrades ושמה בו את ציוני התלמיד. הפונקציה משתמשת ב-initStudent ומחזירה משתנה מטיפוס Student ששדותיו חושבו לפי המחרוזת str ובעזרת הפונקציה calcAvgGrade.

6. כתבו את הפונקציה בעלת החתימה הבאה:

```
void updateGrades (Student* studentArr, int numofStudents, int factor, char factorOp)
```

הפונקציה מקבלת מערך של סטודנטים בשם studentArr ובגודל numofStudents, מספר factor מטיפוס int המציין את גודל הפקטור ותו factorOp הקובע את אופן הפעלת הפקטור בהתאם לנאמר בחלק 4 בקובץ הקלט. הפונקציה מעדכנת את ממוצעי הסטודנטים בהתאם לנאמר בחלק 4 בקובץ הקלט.

7. כתבו את הפונקציה בעלת החתימה הבאה:

```
void freeStudents (Student* studentArr, int numOfStudents)
```

הפונקציה מקבלת מערך של סטודנטים בשם studentArr ובגודל numOfStudents. הפונקציה משחררת את כל הזיכרון שהוקצה במערך studentArr. (שימו לב שלא די בשחרור הזיכרון של מערך הסטודנטים בלבד).

ב- Ex4_q2_123456789.c :

8. כתבו את הפונקציה בעלת החתימה הבאה:

```
void readFromFile (Student* studentArr, int numOfStudents, FILE* fPtrRead)
```

הפונקציה מקבלת מערך ריק של סטודנטים בשם studentArr ובגודל numOfStudents וכן מצביע לקובץ הקלט בשם fPtrRead (שכפי שנראה בסעיף 10 הוא מצביע לתחילת השורה השנייה בקובץ – השורה הראשונה נקראה כבר בmain – הוסבר בנושא "הארגומנטים של התוכנית"). הפונקציה קוראת את שאר הקובץ ובאמצעות הפונקציות שמימשנו במודול הקודם (createWeightsArray, getStudentFromString, updateGrades) היא מעדכנת את מערך הסטודנטים. זכרו לשחרר זיכרון.

9. כתבו את הפונקציה בעלת החתימה הבאה:

```
void printLog (Student* studentArr, int numOfStudents, FILE *fPtrWrite)
```

הפונקציה מקבלת מערך מעודכן של סטודנטים בשם studentArr ובגודל numOfStudents וכן מצביע לקובץ הפלט בשם fPtrWrite. הפונקציה מדפיסה לקובץ בהתאם לדרישות התרגיל.

10. כתבו את פונקציית ה-main, בעלת החתימה הבאה:

```
int main(int argc, char* argv[])
```

הפונקציות האלו של פונקציית ה-main תוארה תחת הנושא "הארגומנטים של התוכנית". עליכם לפתוח את הקבצים לקריאה ולכתיבה. במידה ולפתוח אחד מהקבצים נכשל בפתחה יש להדפיס למסך ה-std (השחור) הודעה מתאימה (`There was error using files\n`) ולצאת עם ערך 1. עליכם להשתמש בפונקציות שמימשנו במודול הקודם (initStudentArray, freeStudents) וכן ב-2 הפונקציות מהסעיפים הקודמים. זכרו גם לסגור את הקבצים ולשחרר זיכרון.

הערות למודול 2:

במודול זה אתם מתבקשים לעבוד עם קבצים ועליכם להשתמש בפונקציות שראינו בהרצאה/בתרגול. פונקציות שימושיות שכדאי לחזור עליהם ולהיות בטוחים שאתם מבינים ומכירים אותם: fopen, fclose, fscanf, fprintf, fgets, fputs, feof.

כמו כן, הערה שיכולה לחסוך לכם הרבה עבודה מיותרת (כולם יודעים זאת אבל הרוב שוכחים): זכרו שכאשר דיברנו בתרגול השני על scanf, אמרנו שניתן לקלוט פורמט כלשהו (% משהו) מתוך מחרוזת. למשל: scanf("VisualStudio%d\n",&year); כאשר year מטיפוס int. אם נקליד למסך השחור VisualStudio2010 ואז enter, אז year יקבל את המספר 2010. (ולא רק להשתמש ב-scanf בצורה scanf("%d",&year)). תוכלו להכליל זאת בקלות ל-fscanf.

כמו כן, ניתן להניח שאורכה המקסימלי של שורה בקובץ הקלט הינו 50. קבעו לכם מספר זה באמצעות define בקובץ ה-h של מודול זה Ex4_q2_123456789.h

מספר הערות חשובות:

הערות כלליות לגבי הקצאת זיכרון:

- בחלק מהפונקציות שרשומות לעיל (`createWeightsArray` , `initStudentArray` , `getStudentFromString`) מבצעים הקצאת זיכרון דינאמי אך לא משחררים אותו, אלא מחזירים משתנה/מערך שמשמש בזיכרון שהוקצה דינאמית. לכן, באחריות התוכנית שקוראת לפונקציות הללו לשחרר את הזיכרון שהוקצה על ידם.
- אם אתם מקצים דינאמית זיכרון נוסף באחת הפונקציות או יותר הרשומות לעיל, אז באחריותכם לשחרר את הזיכרון שהקצאתם דינאמית באותן הפונקציות בהן ביצעתם את הקצאת הזיכרון. במידה וההקצאה של הזיכרון נכשלה יש להדפיס למסך ה-`std` (השחור) הודעה מתאימה ("**Memory allocation failed!\n**") ולצאת עם `exit(1)`. במידה והכל תקין, הפונקציה תחזיר את הערך בו התבקשתם להחזיר באותה הפונקציה.

הערות כלליות לגבי בניית המודולים:

- בקובץ ה-`Ex4_q1_123456789.h` עליכם לשים את כל ה-`includes` , כל ה-`defines` , הגדרת המבנה `Student` כפי שניתנה לעיל וכן את כל ההכרזות של הפונקציות אותן אתם מממשים בקובץ ה-`c`. בשם `Ex4_q1_123456789.c`.
- ב-`Ex4_q1_123456789.c` אין להוסיף שורות קוד מלבד מימושי הפונקציות שלעיל ומלבד השורה "`#include \"Ex4_q1_123456789.h\"`" בתחילת קובץ ה-`c`. כל `include/define` נוספים יש להוסיפם ב-`Ex4_q1_123456789.h` ולא ב-`Ex4_q1_123456789.c`.
- זכרו להוסיף את ה-`include guards` למודול זה (`#ifndef EX4_Q1_123456789_H` בשורה הראשונה, `#define EX4_Q1_123456789_H` בשורה השנייה, `#endif` בשורה האחרונה).
- בקובץ ה-`Ex4_q2_123456789.h` עליכם לשים את כל ה-`includes` , כל ה-`defines` וכל ההכרזות של הפונקציות אותן אתם מממשים בקובץ ה-`c`. בשם `Ex4_q2_123456789.c`.
- ב-`Ex4_q2_123456789.c` אין להוסיף שורות קוד מלבד מימושי הפונקציות שלעיל ומלבד השורה "`#include \"Ex4_q2_123456789.h\"`" בתחילת קובץ ה-`c`. כל `include/define` נוספים יש להוסיפם ב-`Ex4_q2_123456789.h` ולא ב-`Ex4_q2_123456789.c`.
- ב-`Ex4_q2_123456789.h` הוסיפו גם את השורה "`#include \"Ex4_q1_123456789.h\"`" כדי לייבא את המודל שמימשנו. שימו לב שיש לשים שורה זו ב-`Ex4_q2_123456789.h` ולא ב-`Ex4_q2_123456789.c`.
- כאשר בכל מקום שבו נכתב **123456789** יש לכתוב במקום את תעודת הזהות שלכם.

הערות כלליות לגבי ההגשה:

- את כל 4 הקבצים (`Ex4_q1_123456789.c` , `Ex4_q1_123456789.h` , `Ex4_q2_123456789.c` , `Ex4_q2_123456789.h`) יש לשים בתוך קובץ `zip` ולשמור אותו בשם `Ex4_123456789.zip` ואותו עליכם להגיש במודל.
- כאשר יש לכתוב במקום **123456789** את תעודת הזהות שלכם.

בהצלחה!