

סדנת תכנות בשפת C, מס' קורס 67319 - 2018

תרגיל 3

פתרון בעיות נומריות, קריאה מקבצים, מצביעים לפונקציות

תאריך הגשה: 23:55 23/08/2018

הגשה מאוחרת (בהפחתת 10 נקודות): 23:55 24/08/2018

תאריך ההגשה של הבוחן: 23:55 23/08/2018

הנחיות חשובות לכלל התרגילים:

1. בכל התרגילים יש לעמוד בהנחיות הגשת התרגילים וסגנון כתיבת הקוד. שני המסמכים נמצאים באתר הקורס – הניקוד יכלול גם עמידה בדרישות אלו.
2. בכל התרגילים עליכם לכתוב קוד ברור. בכל מקרה בו הקוד שלכם אינו ברור מספיק עליכם להוסיף הערות הסבר בגוף הקוד. יש להקפיד על תיעוד (documentation) הקוד ובפרט תיעוד של כל פונקציה.
3. במידה ואתם משתמשים בעיצוב מיוחד או משהו לא שגורתי, עליכם להוסיף הערות בקוד המסבירות את העיצוב שלכם ומדוע בחרתם בו.
4. עבור כל פונקציה בה אתם משתמשים, עליכם לוודא שאתם מבינים היטב מה הפונקציה עושה גם במקרי קצה (התייחסו לכך בתיעוד). ובפרט עליכם לוודא שהפונקציה הצליחה.
5. בכל התרגילים במידה ויש לכם הארכה, או שאתם מגישים באיחור. חל איסור להגיש קובץ כלשהוא בלינק הרגיל (גם אם לינק overdue טרם נפתח). מי שיגיש קבצים בשני הלינקים מסתכן בהורדת ציון משמעותית.
6. אין להגיש קבצים נוספים על אלו שתדרשו. ובפרט אין להגיש קובץ README אלא אם צוין במפורש שיש צורך בכך (לדוגמא, בתרגיל זה אין צורך להגיש).
7. עליכם לקמפל עם הדגלים `Wall -Wextra -Wvla -std=c99` ולוודא שהתוכנית מתקמפלת ללא אזהרות, **תכנית שמתקמפלת עם אזהרות תגרור הורדה משמעותית בציון התרגיל.** למשל, בכדי ליצור תוכנית מקובץ מקור בשם `ex1.c` יש להריץ את הפקודה:
`gcc -Wextra -Wall -Wvla -std=c11 ex1.c -o ex1`
8. עליכם לוודא שהתרגילים שלכם תקינים ועומדים בכל דרישות הקימפול והריצה במחשבי בית הספר מבוססי מעבדי bit-64 (מחשבי האקווריום, לוי, השרת river). חובה להריץ את התרגיל במחשבי בית הספר לפני ההגשה. (ניתן לוודא שהמחשב עליו אתם עובדים הנו בתצורת bit-64 באמצעות הפקודה `uname -a` ויודא כי הארכיטקטורה היא 64, למשל אם כתוב `x86_64`)
9. לאחר ההגשה, בדקו את הפלט המתקבל בקובץ ה-PDF שנוצר מהpresubmission script בזמן ההגשה. באם ישנן שגיאות, תקנו אותן על מנת שלא לאבד נקודות.
- שימו לב! תרגיל שלא יעבור את הpresubmission script ציונו ירד משמעותית (הציון יתחיל מ-50, ויוכל לרדת) ולא יהיה ניתן לערער על כך.**
10. בדיקת הקוד לפני ההגשה, גם על ידי קריאתו וגם על ידי כתיבת בדיקות אוטומטיות (tests) עבורו היא אחריותכם. בדקו מקרי קצה. במידה וסיפקנו לכם קבצי בדיקה לדוגמא, השימוש בהם יהיה על אחריותכם. **במהלך הבדיקה הקוד שלכם ייבדק מול קלטים נוספים לשם מתן הציון.**
11. הגשה מתוקנת - לאחר מועד הגשת התרגיל ירוצו הבדיקות האוטומטיות ותקבלו פירוט על הטסטים בהם נפלתם. לשם שיפור הציון יהיה ניתן להגיש שוב את התרגיל לאחר תיקוני קוד ולקבל בחזרה חלק מהנקודות - פרטים מלאים נמצאים בפורום ואתר הקורס.

פתרון משוואת החום בשיטת הרלקסציה

במצבים רבים, התפשטות חום דרך חומר מהווה שיקול חשוב בתכנון של מוצר, ניסוי או תהליך. למשל, עבור סיר בישול נרצה חום אחיד בתוך הסיר (על מנת לבשל את האוכל בצורה מיטבית), אך הפרש טמפרטורות גדול בין הסיר לבין הידיות (על מנת שנוכל להחזיק את הסיר). בתרגיל זה תכתבו תוכנה הפותרת את משוואת החום בצורה נומרית עבור מקרים פשוטים (דו-מימדיים). משוואת החום היא משוואה דיפרנציאלית המתארת קשר בין הטמפרטורה בנקודה מסוימת לטמפרטורה בנקודות שסביבה. משוואות מצורה זו ניתנות לפתרון מקורב על ידי [שיטת הרלקסציה](#): נחלק את המרחב לנקודות במרווח קבוע, ובכל סבב של החישוב נעדכן את הטמפרטורה בכל נקודה על ידי הפעלת המשוואה על הנקודות שסביבה. נתונים לכם הקובצים `calculator.h`, `heat_eqn.h` ו-`heat_eqn.c`. אין לשנות קבצים אלו ואין להגיש אותם – הקוד שלכם ייבדק עם גרסאות שונות במקצת של קבצים אלו.

עליכם לכתוב שני קבצים:

1. `calculator.c`

קובץ זה יכיל פונקציית חישוב אשר לה החתימה הבאה:

```
typedef double (*diff_func)(double right, double top, double left, double bottom,
double cell);
double calculate(diff_func function, double ** grid, size_t n, size_t m, point *sources,
size_t num_sources, double terminate, int n_iter, int is_cyclic);
```

הפונקציה תיישם את האלגוריתם הבא:

- חשב את סכום החום ההתחלתי: סכום ערכי התאים במרחב החישוב (`grid` (המערך
 - כל עוד תנאי הסיום לא הושג:
 - הפעל את הפונקציה הנתונה על כל הנקודות במרחב שאינם מקורות
 - חשב את ההפרש בין סכום החום כעת לסכום החום לאחר הסבב הקודם.
 - החזר את הפרש סכום החום של הסיבוב האחרון
- תנאי הסיום הוא: אם `n_iter` גדול מאפס, מריצים את הלולאה כמספר האיטרציות המצויין. אחרת, מריצים את הלולאה עד שההפרש בין סכום החום בסוף האיטרציה הנוכחית לסכום החום בסוף האיטרציה הקודמת קטן מהערך `terminate`.

הפרמטר `is_cyclic` שולט בהתנהגות החישוב על שפת המרחב: אם התקבל הערך `false`, כל נקודה שחורגת מהמערך תזון למשוואה עם ערך 0. אם התקבל ערך `true`, נקודות החורגות מהמערך יחוברו לצד השני בצורה מעגלית. לדוגמא, במערך בגודל 10×10 , השכנים של הנקודה $(0,5)$ יהיו $(1,5)$, $(0,6)$, $(0,4)$ ו- $(9,5)$.

2. `reader.c`

קובץ זה יקבל שם של קובץ משורת הפקודה, יקרא ממנו את הפרמטרים של הבעיה וירייץ את פונקציית החישוב. פורמט הקובץ הוא כדלהלן:

<גודל אזור החישוב (שני מספרים שלמים, n, m)>

<רשימת קואדינטות של מקורות החום (זוגות של מספרים שלמים) וערכן (מספר ממשי)>

<ערך סיום (מספר ממשי בכתוב מדעי)>

<מספר איטרציות בין הדפסות (אפס עבור הדפסה בסוף בלבד)>

<האם מעגלי (0 או 1)>

לדוגמא:

```
50, 50
----
0, 0, 5
0, 1, 5
0, 2, 5
0, 47, -5
0, 48, -5
0, 49, -5
----
1e-3
0
1
```

לאחר קריאת הקובץ, יש להקצות זיכרון בגודל המתאים ולשלוח את הפרמטרים לפונקציית החישוב. בכל פעם שנרצה להדפיס, יש להדפיס לפלט התקני את ערך ההחזרה של הפונקציה calculate ואת התוצאות על פי סדר:

```
<termination value from calculator>
<value of cell 0, 0>, <value of cell 0, 1>, ...<value of cell 0, m>,
<value of cell 1, 0>, <value of cell 1, 1>...
...
```

הערכים מופרדים על ידי פסיק ורווח, ובסוף כל שורה של הלוח מודפסת ירידת שורה. יש להדפיס 2 ספרות מעל ו-4 ספרות מתחת לנקודה העשרונית (ראו את התיעוד של printf).

הנחות מקלות:

- בכל מקרה של תקלה או בעיה, עליכם להדפיס לפלט השגיאות (stderr) הודעה אינפורמטיבית כלשהיא ולצאת בצורה מסודרת עם ערך החזרה שאינו 0, כולל שחרור זיכרון, סגירת קבצים, וכו'.
- שימו לב שאינכם נדרשים לזהות מה הבעיה – מספיק לנסות לקרוא קובץ תקין ואם הקובץ לא עומד בציפיות לוותר ולצאת בצורה מסודרת.
- ניתן להניח כי יש לפחות מקור חום אחד (שורה אחת ברשימת הנקודות).

Makefile

עליכם להוסיף Makefile התומך בפעולות הבאות:

- make - מהדר את הקבצים ומייצר את ex3.exe.
- make all – מייצר את התוכנית ומפעיל אותה על קובץ הקלט הנתון.
- make clean – מסיר את כל הקבצים הזמניים ואת התוכנית הבנויה.

הצגת הנתונים:

להלן קוד פייתון המאפשר להציג את הנתונים לעיל:

```
import matplotlib.pyplot as plt
import csv

with open('output.csv') as file:
    reader = csv.reader(file, quoting=csv.QUOTE_NONNUMERIC)
    n=[]
```

```

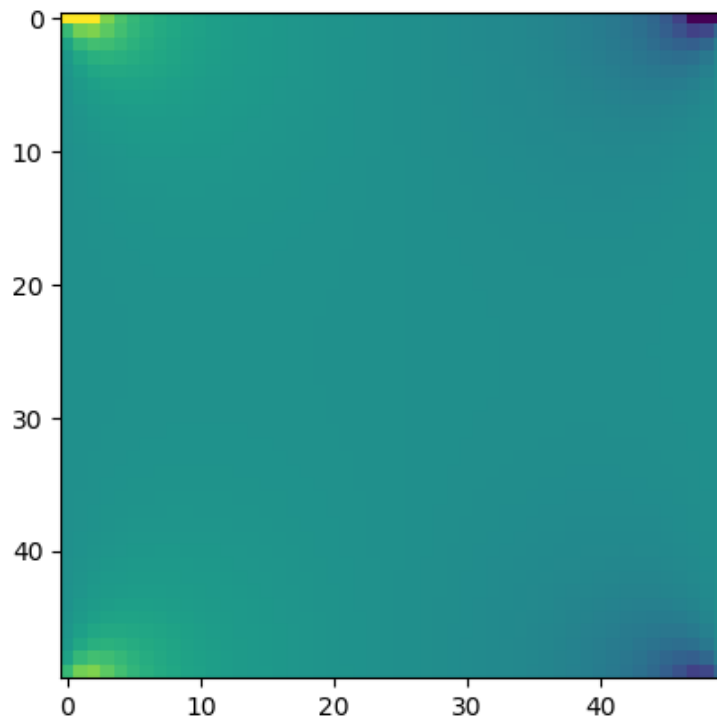
for row in reader:
    if len(row) > 1:
        n.append(row[:-1])
fig = plt.figure()
fig.suptitle("Heat map")
ax = fig.add_subplot(111)
ax.imshow(n)

plt.show()

```

אם פתרתם את התרגיל כראוי, תקבלו תמונה דומה לזו:

Heat map



מידע נוסף (כללי)

1. חל איסור להשתמש במערכים בגודל דינמי (VLA).
2. עליכם לוודא שהקוד שלכם רץ באופן תקין וללא דליפות זכרון. לשם כך עליכם להשתמש בתוכנת `valgrind`.
3. אתם רשאים להשתמש בכל הספריות הסטנדרטיות של C.
4. אתם רשאים (ולעתים אף נדרשים) להגדיר פונקציות נוספות לשימושכם הפנימי.
5. שימו לב שאתם מכירים כל פונקציה בה אתם משתמשים ושאתם בודקים עבור כל פונקציה שהיא הצליחה, ובכלל זאת הקצאות זיכרון.

טיפול בשגיאות:

- הדפסות שגיאה יודפסו אל `stderr`. שימו לב שעל התוכנה לצאת בצורה מסודרת בכל מקרה של בעיה (`exit gracefully`).
- עבור מספר ארגומנטים שגוי יש להדפיס מידע המסביר כיצד להריץ את הקוד (`usage`) ולצאת עם ערך חזרה שונה מ-0.

- מלבד ההנחות הרשומות אין להניח שהקלט תקין - עבור קלט שאינו תקין על התוכנה לצאת באופן מסודר לאחר הדפסת שגיאה אינפורמטיבית כלשהיא (אין פורמט מחייב להודעת השגיאה).

בדיקת התרגיל:

1. התכניות יבדקו גם על סגנון כתיבת הקוד וגם על פונקציונאליות, באמצעות קבצי קלט שונים (תרחישים שונים להרצת התכניות). מכיוון שהחשוב בתרגיל הינו נומרי ונתון לשגיאות עיגול, פתרוןכם ישווה עד כדי טווח שגיאה.
2. אם ישנם מקרים שהוראות התרגיל לא מציינות בבירור כיצד התכנית צריכה להתנהג, הביטו בקבצי הקלט וקבצי הפלט לדוגמה שניתנים לכם ובדקו אם התשובה לשאלתכם נמצאת שם. כמו כן, היעזרו בפתרון בית הספר, הריצו עליו את הטסטים שלכם והשוו להתנהגות תוכניתכם. כמובן שניתן וכדאי להתיעץ בפורום לגבי מקרים שבהם התשובה עדיין אינה ברורה.

חומר עזר:

1. את פתרון בית הספר ניתן למצוא ב:
~plabc/www/ex3/school_sol.tar
2. קבצי בדיקה לדוגמא ניתן למצוא ב:
~plabc/www/ex3/tests_examples.tar
3. רצוי לבצע "בדיקת שפיות" על הפתרונות שלכם לפני ההגשה, באמצעות שרטוט הפתרונות כפי שמתואר בתרגיל.
4. ספריית math של C לשימושכם גם כן.

הגשה:

1. עליכם להגיש קובץ tar בשם ex3.tar המכיל רק את הקבצים הבאים:
 - reader.c
 - calculator.c
 - Makefile
 ניתן ליצור קובץ tar כדרוש על ידי הפקודה:
tar cvf ex1.tar <files to include in tar>
2. לפני ההגשה, פתחו את הקובץ ex1.tar בתיקיה נפרדת וודאו שהקבצים מתקמפלים ללא שגיאות וללא אזהרות. וודאו שההגשה שלכם עוברת את ה-presubmission script ללא שגיאות או אזהרות.
3. אתם יכולים להריץ בעצמכם בדיקה אוטומטית עבור סגנון קידוד בעזרת הפקודה:
~plabc/www/ex1/presubmit_ex3
4. כאשר <directory or file> מוחלף בשם הקובץ אותו אתם רוצים לבדוק או תיקייה שיבדקו כל הקבצים הנמצאים בה (שימו לב שבדיקה אוטומטית זו הינה רק חלק מבדיקות ה-codingStyle).
דאגו לבדוק לאחר ההגשה את קובץ הפלט (submission.pdf) וודאו שההגשה שלכם עוברת את ה-presubmission script ללא שגיאות או אזהרות.

בהצלחה!