

שפת C – תרגיל בית 2

מערכים סטטיים (חד ודו ממדיים), קריאה וכתובה מקבצים, תכנית מכמה קבצי קוד

תאריך הגשה: יום חמישי, 7.8.2014 עד שעה 23:55
תאריך הגשה מאוחרת (בהפחתה של 10 נקודות): יום שישי, 8.8.2014 עד שעה 14:00

שימו לב :

- בתרגיל זה השימוש במערכים הוא רק במערכים סטטיים (ללא הקצאת זיכרון דינמית), בעלי גודל סטטי (גודל המערך קבוע בקוד וידוע כבר בזמן הקומפילציה). שימוש בהקצאת זיכרון דינמית או במערכים בגודל דינמי יוביל לפסילת הסעיף הרלוונטי.
- עליכם לבצע את פקודת הקומפילציה עם הדגל Wall – על מנת לוודא שתכניתכם מתקמפלת ללא אזהרות. תכנית שמתקמפלת עם אזהרות תגרור הורדת נקודות.
- עליכם לבצע את פקודת הקומפילציה עם הדגל Wvla – על מנת לוודא שתכניתכם אכן אינה משתמשת ב VLA – מערכים על המחסנית בעלי גודל שנקבע דינאמית.
- עליכם לוודא שהתרגילים תקינים ושהתכניות רצות על מחשבי בית הספר, במערכות 64-bit (כמו המחשבים שבמעבדות לוי ובאקווריום וכמו השרת river), כי תרגילכם ייבדקו על מערכות אלו וההתנהגות עלולה להיות שונה במערכות אחרות. לפני ההגשה וודאו שהכל עובד על מחשב של בית הספר. ניתן להתחבר מרחוק באמצעות SSH לשרת river.
- עליכם לוודא שבקובץ הפלט submission.pdf המתקבל לאחר ההגשה, אין שגיאות או אזהרות. קראו את מדיניות הקורס אודות הגשת תרגילים שלא עוברים את הסקריפט.
- מיקום הקבצים עבור שתי משימות התכנות :

~slabc/public/ex2/code_files/
~slabc/public/ex2/inputOutput/

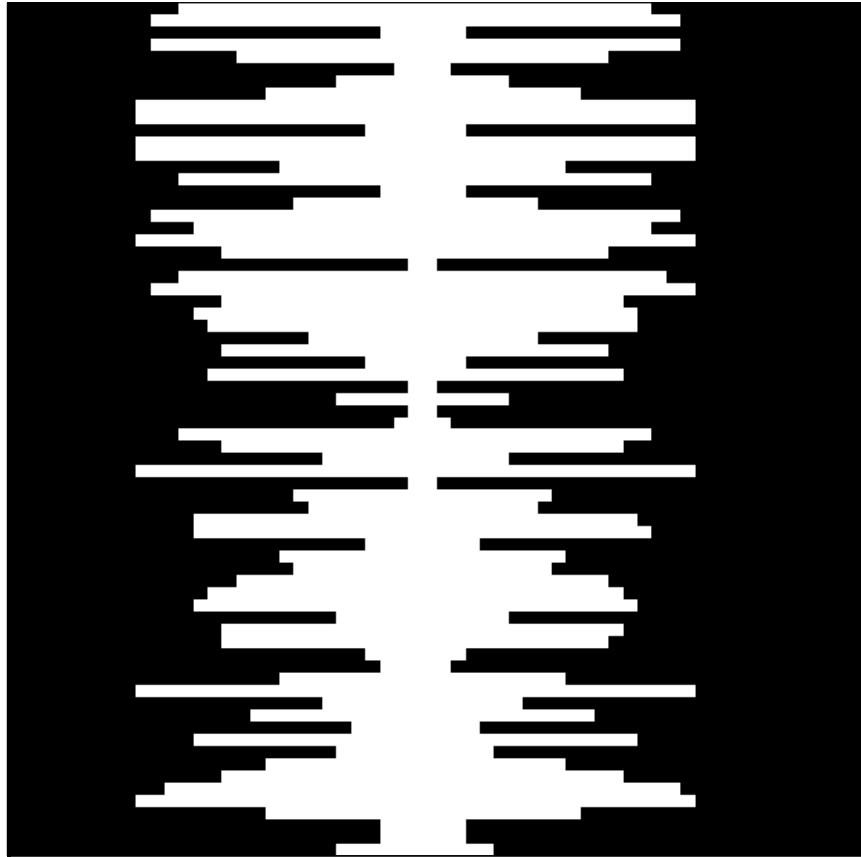
וברשת :

http://www.cs.huji.ac.il/~slabc/ex2/ex2_files.tar

1. משימת תכנות קטנה – שרטוט מערך דו ממדי (10 נקודות) :
לרשותכם קובץ כותר (header-file) בשם `Draw2DCharArray.h`. עליכם לכתוב את הקובץ `Draw2DCharArray.c` ובו לממש את שתי הפונקציות המוגדרות ב `header-file`. שימו לב להגדרות המדויקות של הפונקציות בתיעוד שלהן.
מטרת הקובץ היא הדפסה של מערך דו ממדי של תווים אל המסך.
לרשותכם גם קובץ דרייבר לתכנית הזו: `DrawArrayDriver.c`, שמכיל פונקציה `main`. העזרו בו כדי לבדוק שהמימוש שלכם לשתי הפונקציות נכון. הריצו אותו והתבוננו בקוד. הדרייבר מבקש מהמשתמש לבחור אופציה ולכל אופציה "משרטט ציור" למסך באמצעות הפונקציות שאתם מממשים. עבור האופציה הראשונה מסופק לכם גם קובץ הפלט המתאים (`draw_1.out`). בדקו שאכן הפלט שלכם יוצא זהה בדיוק לקובץ הפלט הניתן. אתם יכולים להוסיף עוד בדיקות לדרייבר כדי לבדוק את מימושכם. אין להגיש את הדרייבר, הוא מיועד רק לצורך בדיקה עצמית.

2. משימת תכנות גדולה – האבנים והבורות:

בכפר קטן (בו גרו יצורים קטנים וכחולים) יש בורות רבים ועגולים. כל בור מתאפיין בכך שהרדיוס שלו משתנה מאוד לאורך עומקו של הבור. לכן בחתך צד נראה שהרדיוס משפיע על רוחבו של הבור:



כל דייר בכפר שרוצה למלא את הבור מקבל מראש הכפר (שמשום מה יש לו כובע אדום) מחסנית של אבנים בצורה של דיסק כאשר כל אבן בעלת רדיוס שונה. מאחר וזו מחסנית של אבנים, ניתן להוציא אותן, ולהפיל אותן לתוך הבור, רק לפי הסדר בו הן נמצאות במחסנית.

מאחר והמשאבים בכפר דלים במיוחד, לא ניתן לזרוק אבן מחוץ לבור אלא כל אבן בראשית המחסנית תיזרק לתוך הבור או שהבור כבר חסום לגמרי. אם אותו דייר יפיל אבן מהמחסנית לתוך הבור, היא תיפול במורד הבור עד שתגיע לאחת משתי נקודות:

1. למקטע בבור שהרדיוס שלו קטן מזה של האבן – אז האבן תשב מעל המקטע הצר מהרדיוס של האבן. לדוגמא, אם מראש הבור ועד עומק 4 מקטעים רדיוס הבור הוא 4 מטרים ואז בעומק 5 מקטעים הרדיוס של הבור הוא 2 מטרים אז אם נזרוק אבן ברדיוס 3 מטרים היא תיפול עד לעומק 4 מקטעים ותתיישב שם, מעל המקטע החמישי.

2. לתחתית הבור – אם האבן לא נתקעה מוקדם יותר ולא היו אבנים אחרות.

עליכם לכתוב תוכנית שתעזור לדיירי הכפר לדעת כמה אבנים הם ידרשו להפיל לתוך הבור כדי לחסום את אותו הבור והאם בכלל יש מספיק אבנים כדי להגיע עד ראש הבור.

הקלט שלכם: לכל דייר בכפר יש בור. בקובץ יופיע מיפוי של הבור שלו (רדיוסים ועומק) ורשימה של הרדיוסים של האבנים שלו לפי הסדר שלהן במחסנית.

הפלט שלכם : יהיה עליכם להדפיס מיפוי של הבור ושל מיקום האבנים לאחר שזרקו אותן לתוך הבור. כמו כן יהיה עליכם לכתוב האם הבור נחסם לגמרי.

התכנית שתבנו מורכבת משלושה שלבים :

- קריאת הנתונים מקבצים בפורמט נתון
- חישוב מספר האבנים שיכנסו לתוך הבור, מיקומם בבור והאם הבור נחסם כולו או לא
- תצוגה למסך של הבור והאבנים שבתוכו

א. קריאת הנתונים :

לרשותכם header-file בשם **RadiusReader.h** ובו שתי פונקציות שעליכם לממש בקובץ **RadiusReader.c**. הפונקציות תשמשנה לשם קריאת קבצי הנתונים של הרדיוסים של הבור ושל האבנים במחסנית.

בהפעלת התכנית שתכתבו, תקבלו בתור פרמטרים שמות של שני קבצים אותם עליכם לקרוא. הקובץ הראשון יכיל את הרדיוסים של הבור. הקובץ השני יכיל את הרדיוסים של האבנים במחסנית. פירוט לגבי אופן הופעת המידע בקבצים נמצא בהמשך. עליכם יהיה להשתמש לפחות באחת משתי הפונקציות בתוכנית הראשית שלכם. עם זאת, אין הכרח להשתמש בשתייהן במהלך המימוש בו אתם בוחרים.

ב. חישוב זריקת האבנים לבור :

עליכם יהיה לממש פונקציה המסוגלת לחשב כמה אבנים ניתן לזרוק מהמחסנית אל תוך הבור. יהיה עליכם לחשב את מספר האבנים שזרקו לתוך הבור ואת מספר המקטעים שנותרו פתוחים בבור. למשל אם האבן הראשונה במחסנית היא גדולה מהרדיוס של ראש הבור אזי התשובה תהיה שנשתמש באבן יחידה ושישארו 0 מקטעים פתוחים בבור. מבחינת מטרת דיירי הכפר, אין משמעות לרדיוס האבן הנמצאת בראש הבור, גם אם היא קטנה משמעותית מרדיוס הבור במקטע העליון מטרם הושגה במידה ומונחת שם אבן כלשהי.

עליכם לממש את האלגוריתם המבצע את החישוב בצורה היעילה ביותר שניתן. עליכם לכתוב בתיעוד הפונקציה את אופן פעולת האלגוריתם ואת דרישות הזיכרון וזמן הריצה שלו במונחים של עומק הבור N וגודל מחסנית האבנים M , במקרה הגרוע ביותר בו אתם משתמשים בכל M האבנים.

אופן מימוש הפונקציה, הפרמטרים אותם היא מקבלת ומחזירה ומיקום המשתנים בהם היא משתמשת (למשל גלובאלי לעומת לוקאלי) נתונים להחלטתכם. על אופן המימוש לשרת את החלק האחרון הכולל הדפסת הנתונים למסך, דבר אותו אתם מתבקשים לבצע בפונקציה נפרדת.

ג. תצוגה למסך של מיפוי הבור והאבנים - חתך צד :

לצורך ההמחשה של תוצאת זריקת האבנים לבור, עליכם לממש את התכנית **PitPluggger.c**. מטרת התכנית תהיה להציג למסך את הבור ואת האבנים שזרקו לתוכו ולתת סיכום של תוצאות זריקת האבנים לבור (כמה אבנים נזרקו, האם הבור חסום וכו').

התכנית תצפה לקבל 2 ארגומנטים בשורת ההפעלה : הראשון – שם של קובץ שמכיל את הרדיוסים של מקטעי הבור, השני – שם של קובץ שמכיל את הרדיוסים של האבנים במחסנית. אם התכנית לא מקבלת 2 ארגומנטים בדיוק, עליה להדפיס את ההודעה :

```
Usage: PitPluggger <Pit Radius input file> <Stone Radius  
input file>\n
```

(ההודעה בשורה אחת ואחריה ירידת שורה). לאחר ההודעה על התכנית להסתיים.

באם כן התקבלו שני הארגומנטים על התכנית להשתמש בשני הקבצים כקלט, זאת אומרת התכנית תצטרך לפתוח את הקבצים לקריאה. אם פתיחת קובץ נכשלת על התכנית להציג הודעת שגיאה ולהסתיים :

```
Unable to open file [filename].\n
```

זיכרו לסגור קובץ פתוח בתום השימוש בו. גם אם התכנית מסתיימת באמצע עקב שגיאה, זיכרו לסגור את כל הקבצים הפתוחים.

איך לפענח את הקבצים (פורמט הקידוד של קבצי הקלט):

הקובץ המכיל את הרדיוסים של מקטעי הבור יהיה בנוי באופן הבא:

- כל שורה תכיל מספר יחיד
- המספרים יהיו מספרים שלמים אי-שליליים בלבד
- השורה הראשונה תכיל את הרדיוס של ראש הבור, השורה השניה של המקטע שמתחתיו וכך הלאה עד לשורה האחרונה המייצגת את הרדיוס במקטע האחרון שלפני תחתית הבור
- ניתן להניח שאין יותר שורות בקובץ מאשר הקבוע MAX_DEPTH אשר מוגדר בקובץ RadiusReader.h

הקובץ המכיל את הרדיוסים של האבנים במחסנית יהיה בנוי באופן הבא:

- כל שורה תכיל מספר יחיד
- המספרים יהיו מספרים שלמים חיוביים בלבד
- השורה הראשונה תכיל את הרדיוס של האבן בראש המחסנית (האבן שניתן להפיל ראשונה אל תוך הבור), השורה השניה את הרדיוס של האבן שניתן להפיל שניה וכך הלאה עד לאבן האחרונה במחסנית.
- ניתן להניח שאין יותר שורות בקובץ מאשר הקבוע MAX_DEPTH אשר מוגדר בקובץ RadiusReader.h

אתם רשאים להניח שבכל קובץ קלט אורך שורה מרבי הוא MAX_LINE_SIZE תווים (המוגדר בקובץ RadiusReader.h). עליכם להסתמך על מידע זה בקריאת שורות מקבצים (קראו את המדריך של הפונקציות fgetc, fscanf).

אתם רשאים להניח שקבצי הקלט יהיו כתובים בפורמט המתואר (אין צורך לוודא תקינות הפורמט של הקלט בתרגיל זה).

פלט התכנית הנדרש:

השורה הראשונה שתודפס למסך תהיה אחת משתי השורות הבאות:

- במידה והצלחתם לחסום את הבור לגמרי. זאת אומרת האבן האחרונה שנזרקה הגיעה למקטע העליון של הבור אז התיישבה מעליו. אז תדפיסו את המשפט הבא (ללא הגרשיים ועם שורה חדשה בסוף):

"Hurrah!! You have successfully plugged that pit ;)\n"

- במידה והבור נותר פתוח, זאת אומרת שלפחות מקטע אחד עדיין חשוף לאחר שזרקתם פנימה את כל האבנים במחסנית, אז תדפיסו את המשפט הבא:

"Oy Vey!! The pit is still open, what will we do now?\n"

השורה השניה שלכם תתאר את עומק הבור הנתון ואת מספר המקטעים מהבור שנותרו חשופים. שורה זו תיכתב בפורמט הבא (ללא גרשיים ועם השורה החדשה בסוף):

"This pit is <INT> levels deep, of which <INT> levels remain open.\n"

כאשר הסימון <INT> יוחלף על ידי המספר השלם המתאים שיודפס כמספר שלם וללא נקודה עשרונית או סימן. לדוגמא אם עומק הבור הוא 12 וחסמנו לגמרי את הבור, עליכם להדפיס את השורה הבאה:

"This pit is 12 levels deep, of which 0 levels remain open.\n"

השתמשו בדוגמאות הקונקרטיות מהקבצים לדוגמא. וודאו שתכניתכם אכן מייצרת פלט הזהה לפלט בקבצים שניתנו לכם.

הערות למשימות התכנות:

התכניות ייבדקו גם על סגנון כתיבת הקוד וגם על פונקציונאליות, באמצעות קבצי קלט שונים (תרחישים שונים להרצת התכניות). הפלט של פתרונותיכם ישווה (השוואת טקסט) לפלט של פתרון בית הספר. לכן עליכם להקפיד על פורמט הדפסה מדויק, כדי למנוע שגיאות מיותרות והורדת נקודות.

לרשותכם כמה קבצי קלט לדוגמא (רק חלק מקבצי הקלט שיהיו בשימוש בבדיקת התרגילים) וקבצי הפלט המתאימים להם (לפי פתרון בית הספר). עליכם לוודא שהתכנית שלכם נותנת את אותו הפלט בדיוק כמו של פתרון בית הספר. מיקומם:

~slabc/public/ex2/inputOutput/

שימו לב למקרי קצה אפשריים מבחינת כמות האבנים במחסנית, כמות האבנים שניתן לזרוק אל הבור, עומק ורדיוס הבור וגודל ההדפסה וכו'.

אם ישנם מקרים שהוראות התרגיל לא מציינות בבירור כיצד התכנית צריכה להתנהג, הביטו בקבצי הקלט וקבצי הפלט לדוגמא שניתנים לכם ובדקו אם התשובה לשאלתכם נמצאת שם. עליכם לכתוב תכנית שתתנהג בדיוק כמו פתרון בית הספר.

שמות קבצי הפלט בנויים בצורה שתאפשר להבין מה היה הקלט לתכנית. לדוגמא, הקובץ
map_pit_1_stone_3.out

מכיל את הפלט של הרצת התכנית ע"י הפקודה:

PitPluggger pit_radius_1.in stone_radius_3.in

Makefile

בכדי להקל לעצמכם על העבודה, הכינו קובץ Makefile עבור התרגיל כבר בשלב מוקדם. זה בהחלט מקל על החיים, במיוחד כשעובדים ועושים הרבה תיקונים.

יש להגיש את קובץ ה Makefile הזה כחלק מהתרגיל.

כמו כן, בכל התרגילים מעתה, עליכם להגיש קובץ Makefile.
על ה-Makefile לתמוך בפקודות הבאות:

- make DrawArrayDriver
המקמפלת את הקובץ DrawArrayDriver.c הנתון לכם ויוצרת קובץ ריצה בשם DrawArrayDriver
- make PitPluggger
היוצרת קובץ ריצה בשם PitPluggger בעזרת הקובץ PitPluggger.c (וקבצים נוספים אם צריך).

- make all
היוצרת את שני קבצי הריצה הנ"ל.

- make tar
היוצר קובץ tar בשם ex2.tar המכיל את כל הקבצים להגשה (ראו מטה).

- make test1
המריץ את המימוש שלכם עם הפרמטרים הבאים:

pit_radius_1.in stone_radius_1.in

ואז מריץ diff על-מנת להשוות את הפלט שלכם לפלט הצפוי.

ראו הערה אודות הפקודה diff בתאור התרגיל הקודם או השתמשו בman כדי להבין כיצד לעשות זאת.

בכתיבת ה-Makefile אתם יכולים להניח שבזמן הקריאה לפקודות השונות קבצי הקוד שאנו סיפקנו לכם יהיו בתיקייה שבה מקמפלים. לא ניתן להניח זאת על קבצי הקלט והפלט. לכן עליכם להתייחס אליהם בעזרת absolute path:

/cs/course/2014/slabc/public/ex2/inputOutput/

שימו לב למקם את ה-targets המתאימים תחת PHONY. (לכל הפחות את יצירת קבצי ההרצה וכן הרצת הבדיקות).

הגשה:

עליכם להגיש קובץ tar בשם ex2.tar שמכיל רק את הקבצים הבאים:

- Draw2DCharArray.c
- RadiusReader.c
- PitPlugger.c
- Makefile

לפני ההגשה, פתחו את הקובץ ex2.tar בתיקה נפרדת וודאו שקבציכם מתקמפלים ללא שגיאות וללא אזהרות.

הריצו את ה script presubmit בצורה הבאה על קובץ ה tar שיצרתם:

```
~slabc/public/ex2/presubmit_ex2 ex2.tar
```

הריצו את התוכנה הבודקת coding style על כל הקבצים בתיקיה מסוימת או על קובץ בודד כך:

```
~slabc/public/codingStyleCheck <file or directory>
```

דאגו לבדוק לאחר ההגשה שקובץ הפלט (submissions.pdf) אינו מכיל שגיאות או אזהרות.

ועכשיו הגיע הזמן לדרדס את התרגיל הזה.

בהצלחה!