

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

סדנאות תכנות בשפת C ו-C++ – קיץ (קורס 67320) C++ – תרגיל 2

תאריך ההגשה של התרגיל והבחון התיאורטי: יום שלישי, ה'10 בספטמבר,
2019 – עד השעה 23:55;
הגשה מאוחרת (בהפחתת 10 נקודות): יום רביעי, ה'11 בספטמבר, 2019 –
עד השעה 23:55.

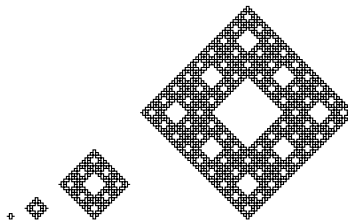
נושאי התרגיל: The rule of 5, המילה השמורה explicit, ירושה ואבסטרקציה.

1 רקע

בתרגיל זה נתנסה בתכונות הירושה של C++ על ידי כתיבת תוכנית המדפיסה עצים פרקטלים.

2 עצים פרקטלים

פרקטל זו צורה גאומטרית המורכבת מעותקים מוקטנים של עצמה. כלומר, מדובר בצורה שחוזרת על עצמה באופן רקורסיבי, כך שאם נבחן חלק מסויים של הפרקטל תחת "זכוכית מגדלת", נגלה שהמראה יהיה זהה בצורתו לפרקטל המקורי. כך, גם אם נבחן את אותו החלק עצמו – נמשיך לראות זאת הלאה והלאה, עד לרמה הראשונה. לפני המשך ההסבר, נראה דוגמה כדי לחדד את האינטואיציה. הפרקטל הבא נקרא Sierpinski carpet, והאיטרציות 1 עד 4 שלו, משמאל לימין, נראות כך:



איור 1: Sierpinski Carpet Orders 1 - 4

הבחינו שככול שאנו מתקדמים באיטרציות, הצורה חוזרת על עצמה בצורה רקורסיבית עוד ועוד. עתה, מהצגנו את הדוגמה – נמשיך. ישנן דוגמאות רבות נוספות לעצים פרקטלים, וחלק מהן נראה בתרגיל זה. לפרקטלים ישנם תכונות מתמטיות המאפיינות אותם. עם זאת, אנו לא נבחן בתרגיל זה את הפן המתמטי של פרקטלים, ונסתפק בכך שנציין שמרבית הפרקטלים מבוססים – כאמור – על התנהגות רקורסיבית. כמו כן, נגדיר את **הממד של הפרקטל**, או **הגובה של הפרטל**, והוא מסמל את דרגת האיטרציה. למשל, ב-Sierpinski Carpet שלעיל, הדרגה של האיור הימני ביותר היא 4, בעוד הדרגה של האיור השמאלי ביותר היא 1.

3 ציור פרקטלים באמצעות תווי ASCII

בתרגיל זה נתרגל תכונות מונחה עצמים, בדגש על פולימורפיזם ואבסטרקציה, בעזרת ציור עצים פרקטלים באמצעות תווי ASCII. התוכנית שנחבר תפעל כך:

1. התוכנית תקרא ותעבד קובץ הכולל הצהרה, או הצהרות, של עצים פרקטלים. תיאור העצים הפרקטלים בהם עליכם לתמוך מוצגת בפרק 4.

2. התוכנית תדפיס את הפרקטלים שבקובץ ההצהרה שנקלט דרך ה-`cli`, אבל **בסדר הפוך** מסדר הופעתם בקובץ.

את פונקציית ה-`main` של התוכנית עליכם ליצור בקובץ `FractalDrawer.cpp` (ושם זה גם יהיה שם קובץ ה-`executable`).

3.1 קלט

התוכנית תקבל ארגומנט יחיד דרך ה-`cli` והוא נתיב לקובץ בפורמט CSV (Comma-separated values)¹. בקובץ, כל שורה תתאר עץ פרקטלי שעלינו לעבד ולהדפיס. באשר לעמודות, אלו יהיו כדלקמן (הסדר משמאל לימין):

תא 0	תא 1
ממד הפרקטל	סוג העץ הפרקטלי

הערות והנחיות באשר לקובץ הפרקטלים:

- על אף שפורמט הקובץ הוא CSV, הסיומת של הקובץ לא חייבת להיות `.csv`. ויכולה להיות כל סיומת אחרת, למשל `.txt` או `.in`.
- **לא ניתן** להניח שהקובץ קיים, עליכם לוודא זאת.
- **לא ניתן** להניח שהקובץ אינו ריק, במקרה שהקובץ ריק – לא יודפס דבר.
- **ניתן** להניח שמדובר בקובץ CSV תקין. **מנגד, לא ניתן** להניח שיהיו רק שתי עמודות – עליכם לוודא זאת. קובץ שאינו עומד בפורמט המתואר **נחשב קובץ פגום** ועליכם לפעול בהתאם להוראות שיוצגו בהמשך (בפרק של טיפול בשגיאות).
- **לא ניתן** להניח שערכי העמודות תקינים. אם אחד מהערכים אינו תקין – מדובר בקובץ פגום. בפרט:

¹ניתן להכיר את הפורמט ולראות דוגמה כאן: https://en.wikipedia.org/wiki/Comma-separated_values

- אם התבקש עץ פרקטלי שלא קיים בהתאם לפרק 4 לתרגיל, מדובר בקובץ פגום.
- אם ממד העץ קטן ממש מ-1, מדובר בשגיאה (כלומר, בהינתן ש- n הוא ממד העץ, אזי מתחייב ש- $n > 0$).
- אם גובה העץ גדול ממש מ-6, מדובר בשגיאה (כלומר, בהינתן ש- h הוא גובה העץ, מתחייב ש- $0 < h \leq 6$).

- הנכם **רשאים** להשתמש בספריית filesystem² של boost³ (ובה **בלבד**).
- הנכם **רשאים** לעשות שימוש ב-STL (C++ Standard Template Library).

3.2 ציור הפרקטלים

לאחר קריאת הקובץ, עליכם לצייר את הפרקטלים שהתבקשו בקובץ **בסדר הפוך לסדר הופעתם**. כפי שנוכחתם לראות מהגדרת קובץ ה-CSV, על המשתמש לבחור מבין מספר פרקטלים אפשריים – כולם מתוארים בפרק 4 לתרגיל. את הפרקטלים תציירו בעזרת ASCII Art, כפי שיוסבר בהמשך. מכאן שאנחנו עובדים עם `std::string` ו-`char*`. שימו לב לדגשים וההנחיות הבאות:

- בתרגיל זה הנכם רשאים לבחור כיצד לעצב את התוכנה, כך שתוכלו לעצב את מחלקותיכם כראות עיניכם.
- את כל המחלקות הנוגעות לציור פרקטלים עליכם לשמור ב-`Fractal.h`, `Fractal.cpp`.
- **שימו ⚠:** עבדו לפי כללי תכנות מונחה עצמים שלמדתם בקורס זה ובייחוד בקורס מבוא לתכנות מונחה עצמים. חלק **משמעותי מאוד** מהניקוד שניתן לתרגיל הוא עבור עיצוב נכון וראוי, ושימוש בירושה.
- זכרו לעשות שימוש ב-`The rule of 5` ובשאר הכללים שנלמדו בקורס באשר להצהרה על מחלקות ב-C++.
- **לעניין הפלט, שימו לב:** לאחר הדפסת כל פרקטל, **לרבות** האחרון שמופיע בקובץ, ש להדפיס שורה אחת ריקה.

3.3 רהתמודדות עם שגיאות

עליכם לטפל במקרים בהם לא התקבל קלט תקין. אם מספר הארגומנטים שנשלחו לתוכנה אינו תקין, עליכם להדפיס ל-`stderr` את הפלט:

```
Usage: FractalDrawer <file path>\n
```

מנגד, אם נתקלתם בקלט שגוי – דהיינו קובץ שאינו קיים או בקובץ פגום, עליכם להדפיס:

```
Invalid input\n
```

בשני המקרים "`\n`" מסמן ירידת שורה (כפי שנהוג להדפיסה ב-C++). לאחר הדפסת הפלט, בשני המקרים עליכם לסגור באופן מיידי את התוכנית עם קוד סיום `EXIT_FAILURE`.

²ראו: https://www.boost.org/doc/libs/1_70_0/libs/filesystem/doc/index.htm
³תוכלו לקרוא על הספרייה המופלאה הזו כאן: [https://en.wikipedia.org/wiki/Boost_\(C%2B%2B_libraries\)](https://en.wikipedia.org/wiki/Boost_(C%2B%2B_libraries))

4 פרקטלים

להלן הפרקטלים בהם על תוכניתכם לתמוך. טרם ניגש לכך, שימו לב לדגשים הבאים:

- **מספר תת הסעיף יהיה מספר הפרקטל שעל המשתמש להזין.** לכן, למשל, Sierpinski Carpet מופיע בסעיף 4.1 ולכן כדי להדפיסו על המשתמש לבקש את מספר הפרקטל 1. לכן $fractal \in \{1, 2, 3\}$.
- נרצה להדפיס את הפרקטלים ב-ASCII Art, לכן נבחר את התווים בהם נשתמש:
 - כל פעם שנרצה לצייר חלק מהפרקטל, נשתמש בתו **סולמית** (“#”).
 - כל פעם שנרצה לדלג ולא להדפיס, נשתמש ב**רווח יחיד**.
- להלן תראו שנתאר כל עץ פרקטלי בצורה אינדוקטיבית, על $n \in \mathbb{N} \cup \{0\}$. עם זאת, בקובץ המתאר את רשימת העצים הפרקטלים שעל התוכנית להדפיס – לא נאפשר “לגשת” למקרה הבסיס (בו $n = 0$), ומקרים אלו יחשבו כקובץ פגום.

4.1 Sierpinski Carpet

זהו פרקטל שתואר לראשונה על ידי Waław Sierpiński בשנת 1916. נתאר את בניית הפרקטל בצורה אינדוקטיבית על $n \in \mathbb{N} \cup \{0\}$:

- **בסיס:** כאשר $n = 0$, נתחיל בסימון יחיד (כלומר ריבוע אחד קטן, שאצלינו, כאמור, מסומן ב-#).
- **צעד:** בכל שלב ניצור 9 ריבועי משנה, שניתן לתאר בטבלה הבאה:

#	#	#
#		#
#	#	#

אם כך, במקרה הזה בכל שלב רקורסיבי, נוצר grid מסדר 3×3 בו **מציגים את תת-הפרקטל בסדר $n - 1$ אם ורק אם הוא לא תת-הפרקטל האמצעי** (כלומר זה שבמיקום (2, 2)). לדוגמה, בעוד כאשר הממד הוא 0 יודפס רק “#”, כאשר הממד הוא 1, נקבל:

```
###
# #
###
```

וכאשר הממד הוא 2 נקבל:

```
#####
# ## ## #
#####
###   ###
# #   # #
###   ###
#####
# ## ## #
#####
```

Sierpinski Sieve 4.2

בהמשך לסעיף 1.4, גם זה פרקטל הקרוי על שמו של Waław Sierpiński. נתאר את בנייתו באופן אינדוקטיבי על $n \in \mathbb{N} \cup \{0\}$:

- **בסיס:** כאשר $n = 0$, נתחיל בסימון יחיד (כלומר ריבוע אחד קטן, שאצלינו, כאמור, מסומן ב-#).

- **צעד:** בכל שלב ניצור 4 ריבועי משנה, שניתן לתאר בטבלה הבאה:

#	#
#	

Cantor Dust 4.3

פרקטל זה מבוסס על אחת מהצורות הגיאומטריות של Cantor ונוכל לתארו באופן אינדוקטיבי על $n \in \mathbb{N} \cup \{0\}$ כך:

- **בסיס:** כאשר $n = 0$, נתחיל בסימון # יחיד.

- **צעד:** בכל שלב ניצור 4 ריבועי משנה, שניתן לתאר בטבלה הבאה:

#		#
#		#

5 דוגמה

מאחר שמדובר בהדפסת ASCII לא נכלול בתיאור התרגיל דוגמה מורכבת. מומלץ מאוד לצפות בפתרון בית הספר. עם זאת, בקצרה, אם נניח ש-f.txt מכיל את התוכן:

1,1

1,2

כלומר f.txt מזהיר על שני פרקטלים – Sierpinski Carpet מסדר 1 ומסדר 2. נקבל:

```
$ FractalDrawer f.txt
```

```
#####
```

```
# ## ## #
```

```
#####
```

```
###   ###
```

```
# #   # #
```

```
###   ###
```

```
#####
```

```
# ## ## #
```

```
#####
```

```
###
```

```
# #
```

```
###
```

אשר שורה שנפתחת ב-\$ מסמנת את הפקודה שהוקלדה. שימו לב לשורה הריקה בין הפרקטלים ובסוף הפלט.

6 נהלי הגשה

- קראו בקפידה את הוראות תרגיל זה ואת ההנחיות להגשת תרגילים שבאתר הקורס.
- **שימו לב:** מאחר שמדובר בתרגיל הבוחן, בין היתר, עיצוב תוכנה – הבדיקה הידנית “המדגמית” תהא רחבת היקף, ונבחר לבדוק תרגילים רבים יותר **באופן משמעותי**. אנא הקפידו לעצב את התוכנית כהלכה ובהתאם לעקרונות שנלמדו. ציון תרגילים שלא יממשו עקרונות OOP כהלכה יפגע **באופן משמעותי מאוד**, גם אם הם עומדים במלוא הבדיקות האוטומטיות.
- **שימו לב:** מאחר שתוכניתכם מדפיסה ערכי ASCII ונבדקת באופן אוטומטי, עליכם לדייק בהדפסה. זכרו להשתמש אך ורק בתו “#” להדפסת הפרקטלים, ובתו רווח (יחיד) לתא ריק ב-grid. בסופו של דבר, עבור כל פרקטל, תצטרכו להדפיס כמות תווים הזוהה לכמות התווים ב-grid. לכן, למשל, עבור Sierpinski Carpet מממד 1, הכולל grid של 3×3 , נקבל הדפסה של 9 תווים ועוד 3 ירידות שורה.
- זכרו שעליכם לקמפל את התוכנית כנגד מהדר לשפת C++ בתקן שנקבע בקורס. כמו כן, זכרו שעליכם **לתעדף** פונקציות ותכונות של C++ על פני אלו של C. למשל, נעדיף להשתמש ב-new ו-delete על פני malloc ו-free, וכן נעדיף להשתמש ב-std::string מאשר ב-char*.
- **נזכיר:** כאמור בהנחיות הכלליות להגשת תרגילים – הקצאת זיכרון דינמית **מחייבת** את שחרור הזיכרון, למעט במקרים בהם ישנה שגיאה המחייבת סגירת התוכנית באופן מיידי עם קוד שגיאה (כלומר קוד יציאה השונה מ-0). תוכלו להיעזר בתוכנה valgrind כדי לחפש דליפות זיכרון בתוכנית שכתבתם.
- פתרון בית הספר זמין בנתיב

~proglab/www/cpp_ex2/FractalDrawer

- בתרגיל זה אין להגיש קובץ Makefile. עליכם ליצור קובץ tar הכולל **אך ורק** את הקבצים FractalDrawer.cpp, Fractal.h, Fractal.cpp, README. ניתן ליצור קובץ tar כדרוש על ידי הפקודה:

```
$ tar -cvf cpp_ex2.tar FractalDrawer.cpp Fractal.cpp  
Fractal.h README
```

שימו לב: קבצי קוד המקור שתכתבו נדרשים להתקמפל כהלכה עם `std++14`, כנדרש בהוראות להגשת תרגילים שפורסמו באתר הקורס.

אם בחרתם להשתמש ב-boost, עליכם לוודא שהגרסה שעמה עבדתם נתמכת במחשבי בית הספר.

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

~proglab/www/cpp_ex2/presubmission

בהצלחה!!