

השלמות תכנות מונחה עצמים סמסטר א' תשפ"ג מועד א'

- אין להגיש למערכת הבדיקות קוד עם הדפסות דיבאג.
- מותר להגיש כמה פעמים שתמצאו במוד אימון ובמוד הגשה.
- לפני תום הבחינה עליכם להגיש במוד הגשה ואז במוד הגשה סופית.
 - הגשה סופית נועלת את הציון האחרון במוד הגשה, ואינה מחוללת בדיקה.

שאלה 1 (50 נק') נבדקת אוטומטית בלבד.

בקובץ Q1.h ישנה המחלקה SnakesNladders שצריכה לייצג משחק של נחשים וסולמות באופן הבא:

הלוח הוא בגודל 10x10 כאשר בכל משבצת יש אינדקס. הספירה מתחילה מ 0 ומסתיימת ב 99 כמו בתמונה מימין.

השחקנים מתחילים במשבצת 0, ומטילים קובייה שמורה להם כמה משבצות להתקדם. לדוגמה אם היו במשבצת 8 וקבלו 3, אז יגיעו למשבצת 11.

אם יש במשבצת סולם, אז הוא יוביל את השחקן לאינדקס גבוה יותר, למשל מ 27 ל 36 או מ 53 ל 86 כפי שמופיע בתמונה מימין.

אם יש במשבצת ראש נחש, אז הוא יוביל את השחקן לאינדקס נמוך יותר, למשל מ 31 ל 14 או מ 89 ל 56 כפי שמופיע בתמונה מימין.

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89
90	91	92	93	94	95	96	97	98	99

עליכם לממש את המחלקה SnakesNladders באופן הבא:

- תייצגו את המידע של הלוח באיזה מבנה נתונים שנוח לכם.
- המתודה addLadder תקבל שני אינדקסים from, to ואם אכן ה from קטן מה to וה to בתוך גבולות הלוח אז יוגדר סולם מאינדקס from לאינדקס to.
- המתודה addSnake תקבל שני אינדקסים from, to ואם אכן ה to קטן מה from וה from בתוך גבולות הלוח אז יוגדר נחש מאינדקס from לאינדקס to.
- מחלקה פנימית iterator תגדיר איטרטור למשחק.
 - האופרטור * יחזיר את האינדקס עליו מצביע האיטרטור
 - האופרטור ++ יקדם את האיטרטור לאינדקס הבא (אם הוא בגבולות הלוח)
 - האיטרטור ימשיך לטייל אוטומטית על הלוח כל עוד הוא מובל ע"י סולמות או נחשים.
 - הוא יעצר על משבצת ללא תחתית של סולם או ראש של נחש.

חוזרים למחלקה החיצונית SnakesNladders

- המתודה getIterator תחזיר איטרטור חדש המצביע על משבצת 0.
- המתודה hasLoops תחזיר אמת או שקר בהתאמה לקיומן של לולאות בלוח.
 - למשל אם יש לי סולם מ 1 ל 10, וסולם מ 10 ל 39, ונחש מ 39 ל 1 אז קיימת לולאה.

שאלה זו נבדקת באופן אוטומטי בלבד ולכן עליכם להקפיד להגיש קוד מתקמפל, ללא שגיאות ריצה וללא לולאות אינסופיות...

שאלה 2 (50 נק') נבדקת אוטומטית בלבד

בקובץ Q2.h עליכם לממש את הבאים:

- המחלקה Shape מתארת "צורה" מופשטת. תוכלו להוסיף לה מתודות ע"פ הצורך.
- המחלקה Point היא סוג של צורה המתארת נקודה דו-ממדית ע"י שני int-ים פומביים x,y.
- המחלקה Line היא סוג של צורה המתארת קו ע"י שתי נקודות פומביות from, to.
- המחלקה Area היא סוג של צורה המתארת שטח ע"י שני קווים פומביים l1,l2.

לכל המחלקות לעיל עליכם להוסיף בנאים ומתודות ע"פ הצורך כדי לתמוך בשתי הפונקציות הבאות שעליכם לממש:

- הפונקציה save – בהינתן מערך של Shape**, גודלו ושם קובץ, היא תשמור בקובץ את כל הצורות שבמערך. אתם מחליטים על פורמט הקובץ, חישבו מה נדרש לשמור כדי לתמוך בטעינה מהקובץ.
- הפונקציה load – בהינתן שם של קובץ היא תטען מתוכו את כל האובייקטים של הצורות השונות ותחזיר מצביע מסוג Shape** למערך שמכיל את הצורות הללו.

הבדיקה במוד האימון זהה לבדיקה במוד ההגשה. הבדיקה מייצרת כמה אובייקטים עם ערכים אקראיים, שומרת אותם באמצעות save וטוענת אותם באמצעות load. הבדיקה משווה שאכן קבלנו מהטעינה בדיוק את אותו המידע ששמרנו.

הערה: אין באמת צורך להשתמש ב typeid, אבל אם תבחרו להשתמש בו זכרו של typeid().name() מעל windows נקבל את המילה class, רווח ואז את שם המחלקה, למשל "class Line". לעומת זאת מעל לינוקס במערכת הבדיקות נקבל את מס' האותיות ואת שם המחלקה, למשל "4Line".

שאלה זו נבדקת באופן אוטומטי בלבד ולכן עליכם להקפיד להגיש קוד מתקמפל, ללא שגיאות ריצה וללא לולאות אינסופיות...

הגשה

עליכם להגיש את Q1.h, Q2.h למערכת ההגשה במערכת הבדיקות למבחנים בעמוד המודל של הקורס או בכתובת <https://cktest.cs.colman.ac.il/> בקורס OOP מועד א'.

בכל הגשה יש להגיש את כל הקבצים ורק אותם.

תוכלו להגיש כמה פעמים שתמצאו במוד אימון ובמוד הגשה.

לפני תום הבחינה עליכם להגיש את הבחינה **במוד הגשה** ואז במוד **הגשה סופית**.

מוד הגשה סופית לא מבצע בדיקה, אלא נועל את הציון האחרון של מוד הגשה.

בהצלחה!