

## תכן תנועת רובוטים וניווט מבוסס חיישנים - 036044

### מיני פרויקט – חלק 1

מגיש: יעקב וקסמן

ת.ז.: 316153261

מגיש: אלמוג אלבאום

ת.ז.: 313269839

### שאלה 1

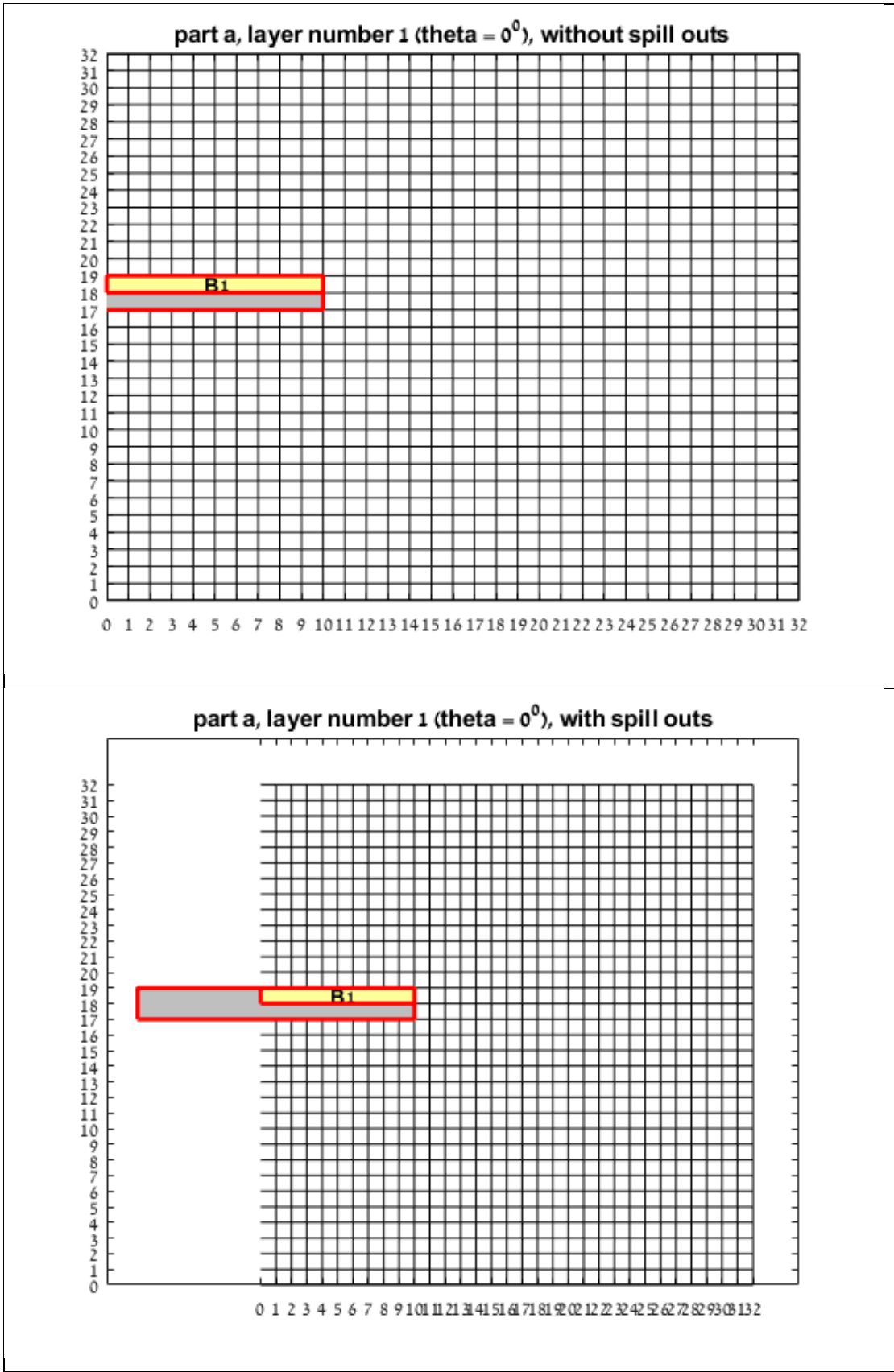
1. Write a program that implements the algorithm as follows:

- **Input:** Description of a convex polygonal robot and a single convex polygonal obstacle, as a list of their vertices described in counter-clockwise order.
- **Output:** For 32 regularly spaced  $\theta$  layers (starting at  $\theta = 0$  until  $\theta = 2\pi - 2\pi/32$ ), compute the vertices of the c-obstacle slices, and draw the lines connecting these vertices.
- **Test:** Test your code on the robot  $A$  and the obstacle  $B_1$  given in Figure 1. Give us a copy of your code and a printout of layers 1, 8, 16, 32.

הקוד המצורף מורכב משש פונקציות בהן אנו משתמשים לפתרון הבעיה:

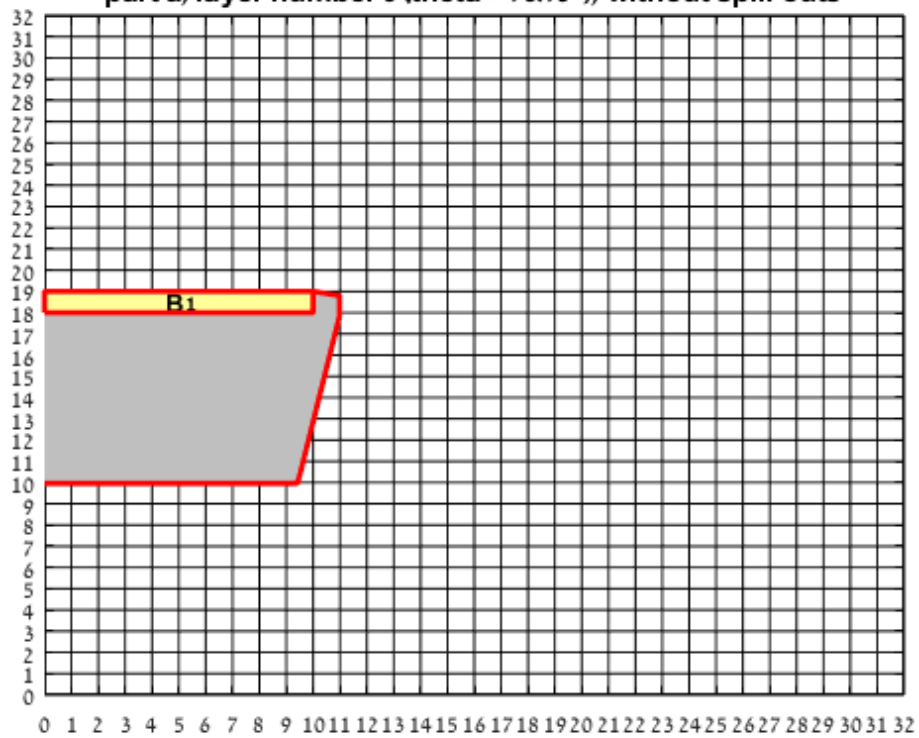
1. **Main** – זו הפונקציה הראשית. במהלך הרצת הפונקציה מבוצעת קריאה ליתר הפונקציות המשמשות לפתרון כלל סעיפי הפרויקט.
2. **Obstacles** – הפונקציה מכילה את נתוני המכשולים שבחדר. כל מכשול מיוצג על ידי מטריצה כאשר כל עמודה מייצגת קודקוד של המכשול כך שהשורה הראשונה מתארת את המיקום בציר x והשורה השנייה את המיקום בציר y. אופן ייצוג זה ימומש גם עבור קודקודי הרובוט ו-c-obstacle בהמשך.
3. **Robot in orientation** – הפונקציה מכילה את מיקום קודקודי הרובוט כאשר אחת הנקודות מוגדרת כראשית הצירים והיתר משתנות לפי האוריינטציה.
4. **Calc CB** – הפונקציה אותה נדרשנו לממש בסעיף זה.
5. **Plot room and CB** – הפונקציה בה נעשה שימוש לצורך הדפסת החדר והמכשולים. הפונקציה מאפשר להגביל את התצוגה לממדי החדר או לא להגביל את התצוגה ובכך לזהות זליגה של המכשולים (במידה וקיימים).  
יצוין שכל מכשול ו-c-obstacle המתאים לו מוקפים בצבע זהה וייחודי ביחס לישר המכשולים.
6. **OcuppencyGrid** – פונקציה אשר יוצרת מפה דיסקרטית (מערך דו ממדי) עבור חתך  $\theta$ . כל תא אשר חלק ממנו שייך למרחב הקונפיגורציה מקבל ערך 1.

להלן הפלט המתקבל עבור הרצת שכבות [1,8,16,32] סביב מכשול B1:

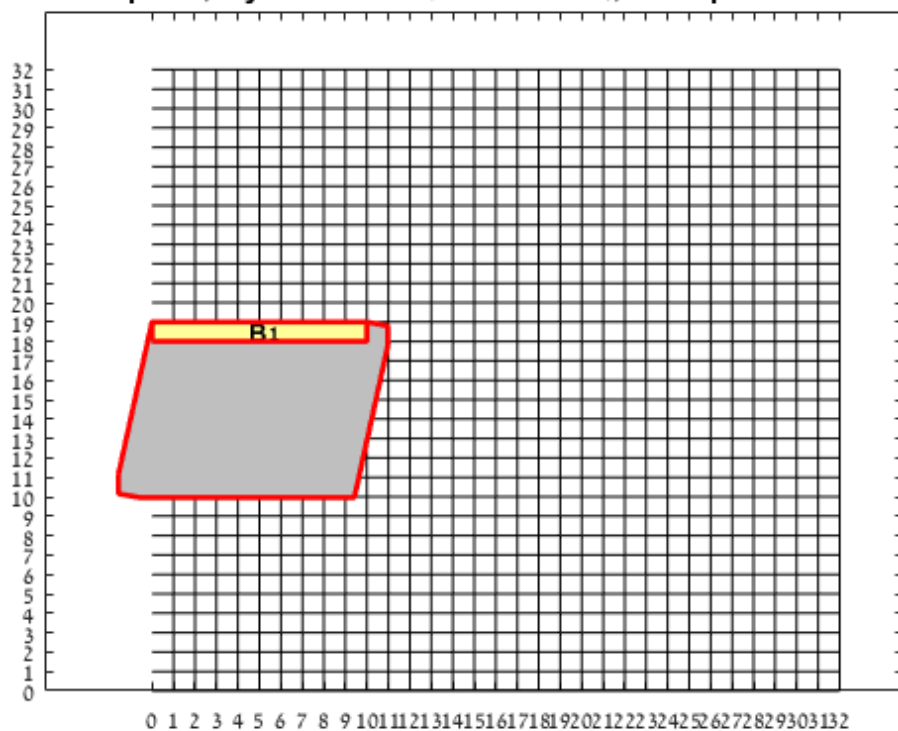


ניתן לראות שעבור השכבה הראשונה התקבלה זליגה.

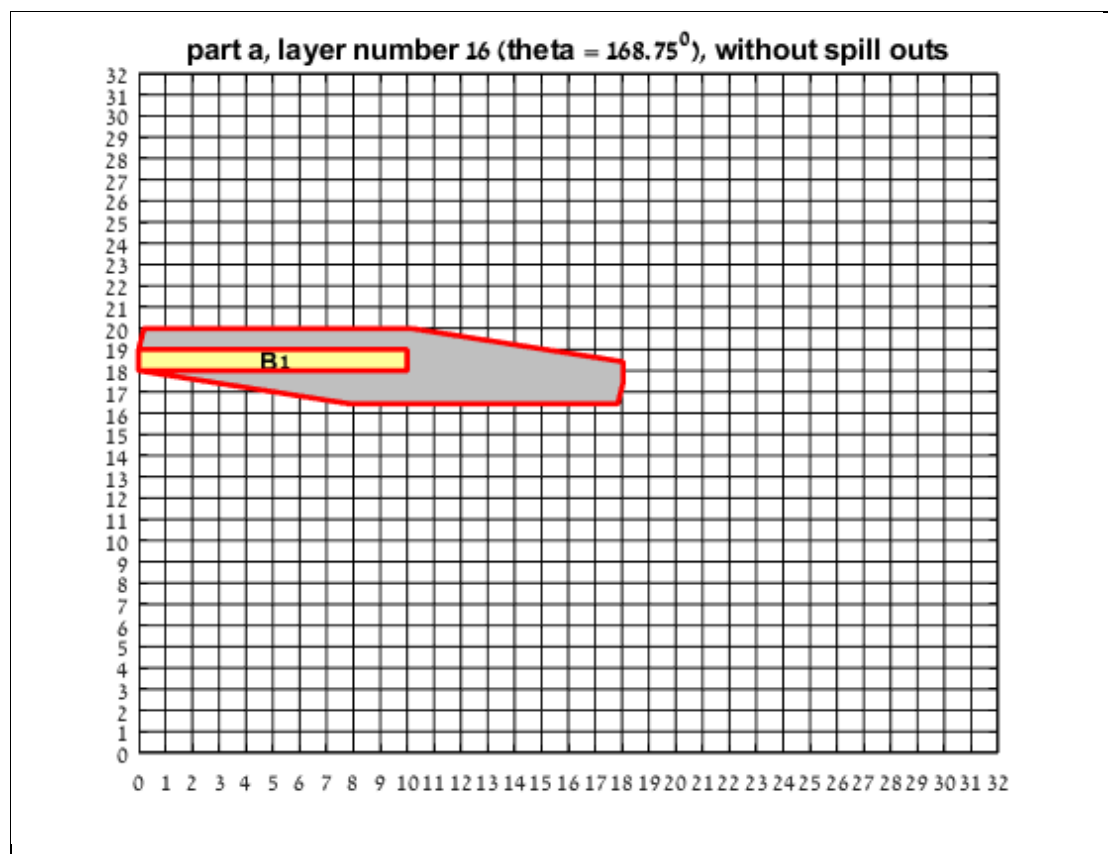
part a, layer number 8 ( $\theta = 78.75^\circ$ ), without spill outs



part a, layer number 8 ( $\theta = 78.75^\circ$ ), with spill outs

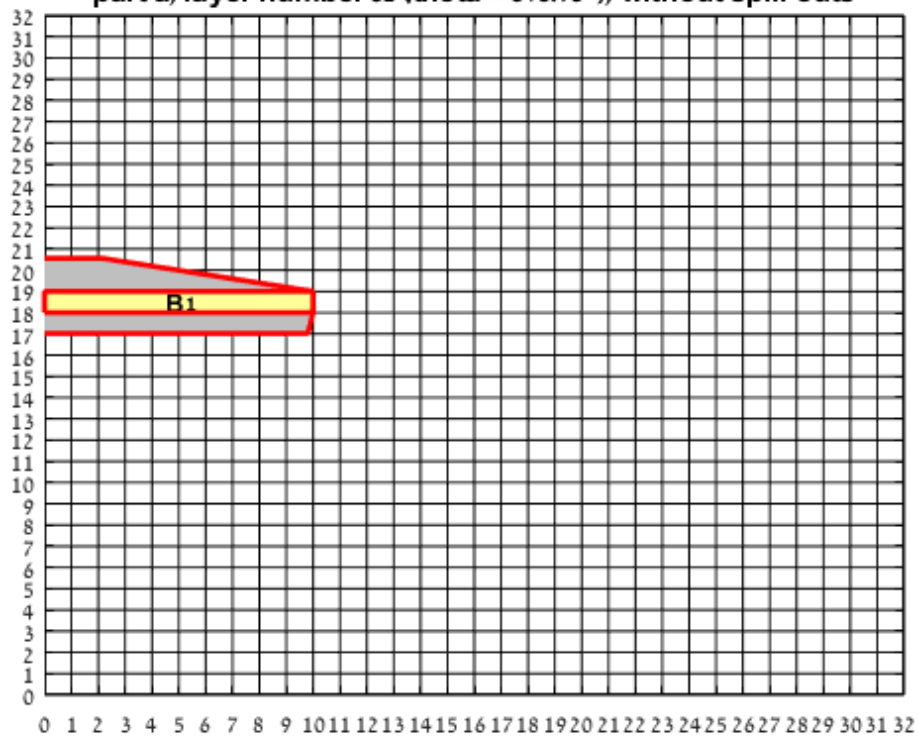


ניתן לראות שעבור שכבה 8 התקבלה זליגה.

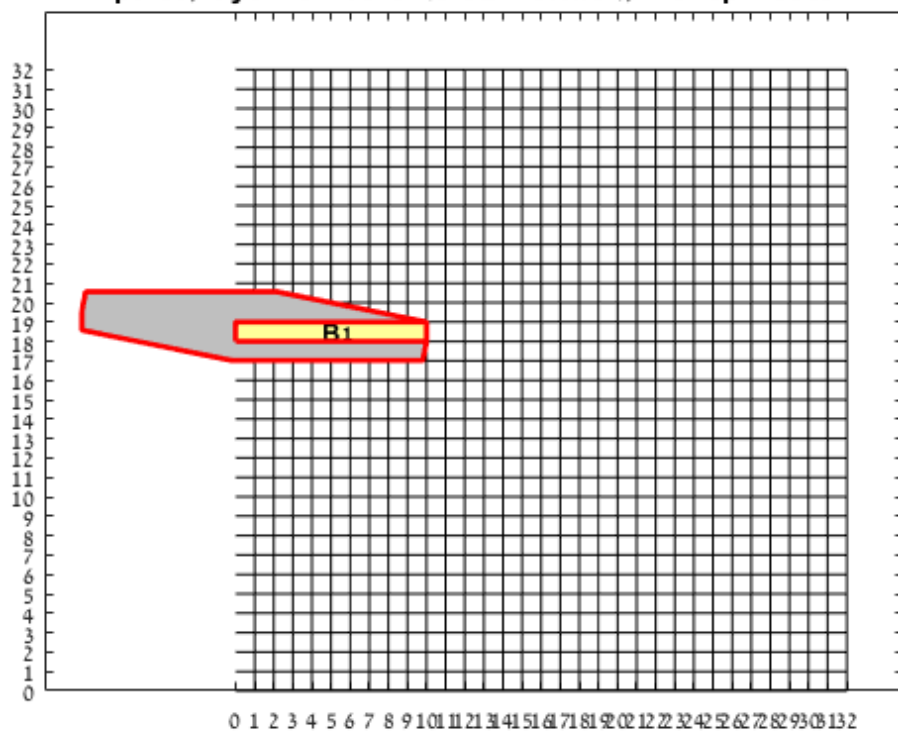


ניתן לראות שעבור שכבה 16 לא התקבלה זליגה.

part a, layer number 32 ( $\theta = 348.75^\circ$ ), without spill outs



part a, layer number 32 ( $\theta = 348.75^\circ$ ), with spill outs

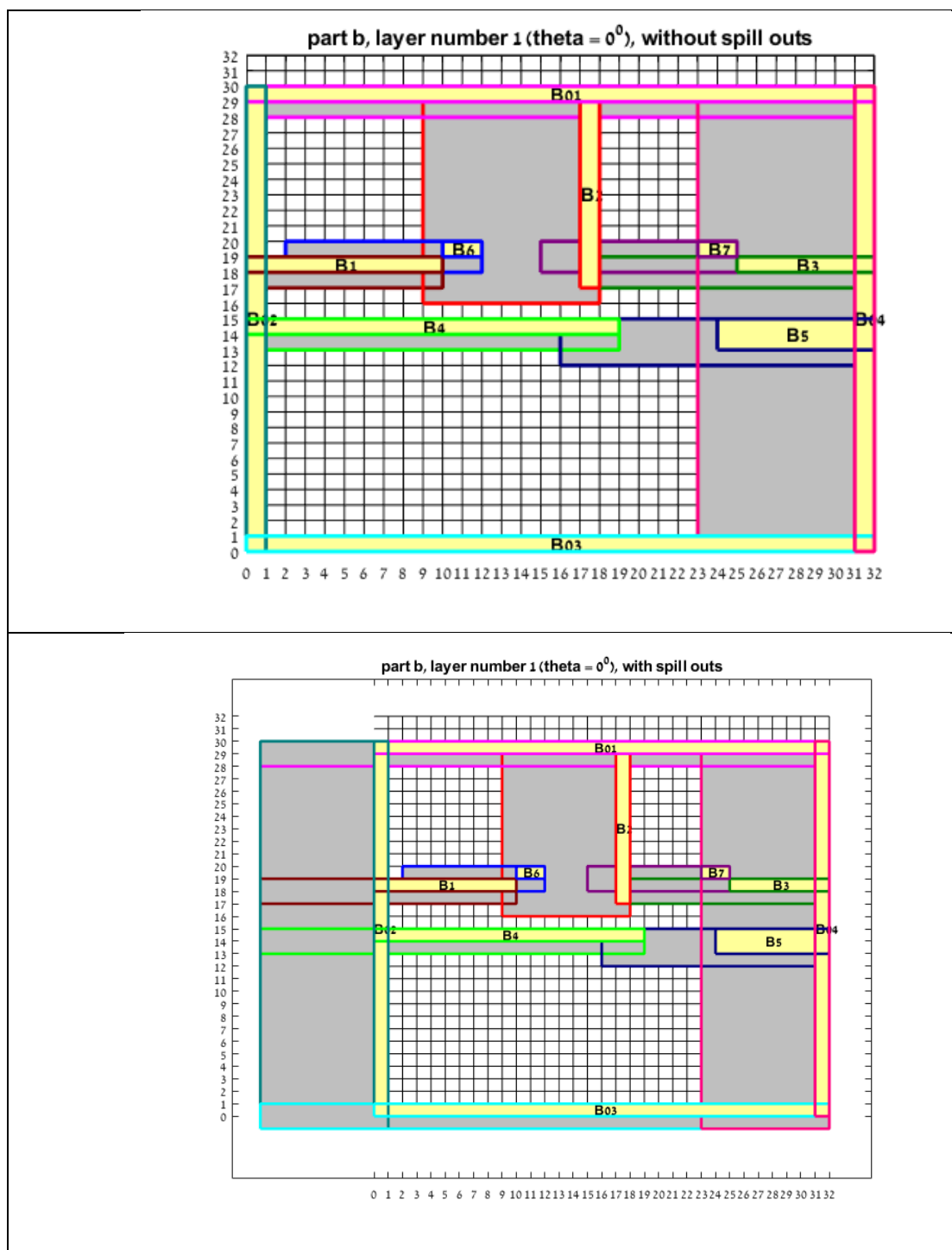


ניתן לראות שעבור שכבה 32 התקבלה זליגה.

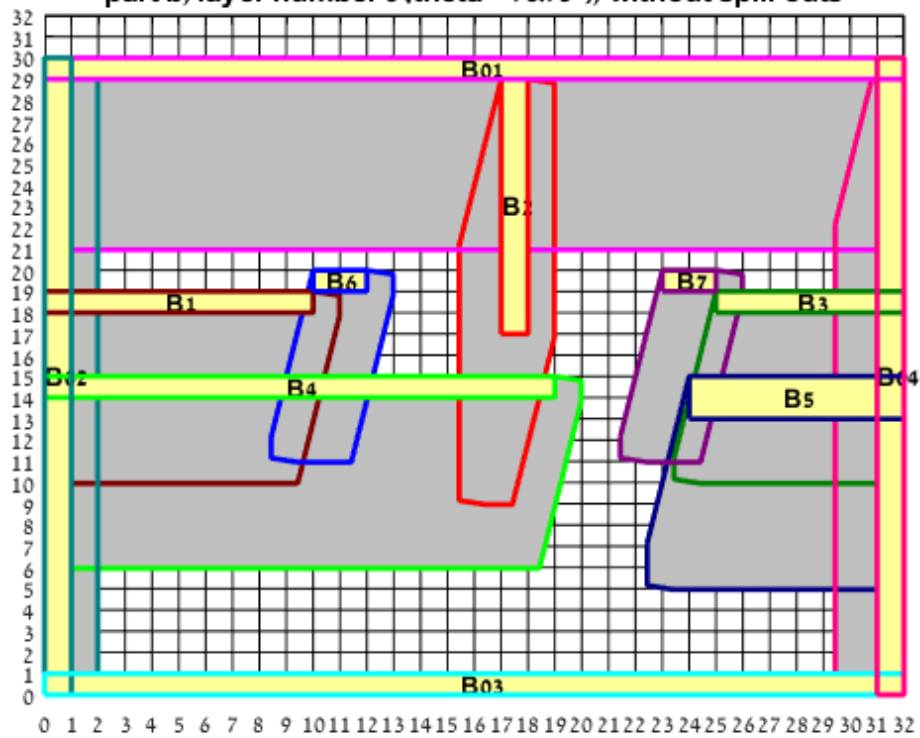
## שאלה 2

2. Extend your code to accommodate a list of possibly overlapping convex polygonal obstacles. Test your code on the example of Figure 1. The example shows an apartment with two bedrooms and a living room connected by a short corridor. The outer wall is represented by a union of four rectangular obstacles  $B_{01}, B_{02}, B_{03}, B_{04}$ . The interior walls are rectangular obstacles  $B_1, \dots, B_5$ . The two doors are  $B_6, B_7$ . Coordinates of  $B_2$ : (17, 17), (18, 17), (18, 29), (17, 29). Coordinates of  $B_{01}$ : (0, 29), (32, 29), (32, 30), (0, 30). Use the figure to build the input. Give us a *printout of layers* 1, 8, 16, 32. It would be nice if you mark in each slice the regions belonging to the respective physical obstacles.

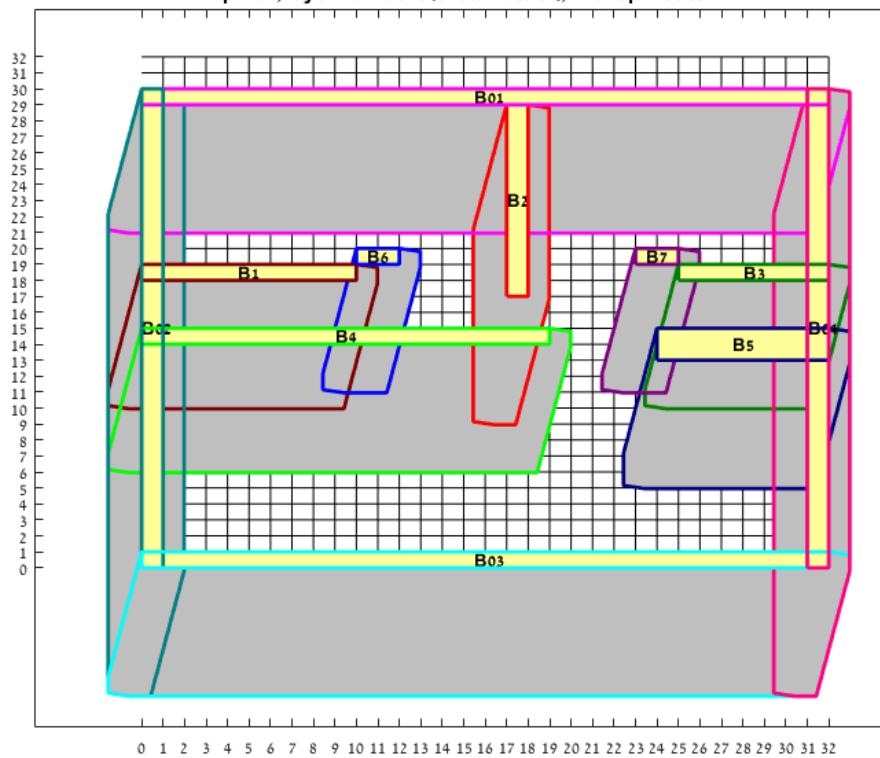
להלן הפלט המתקבל עבור הרצת שכבות [1,8,16,32] סביב מכשול כלל המכשולים :



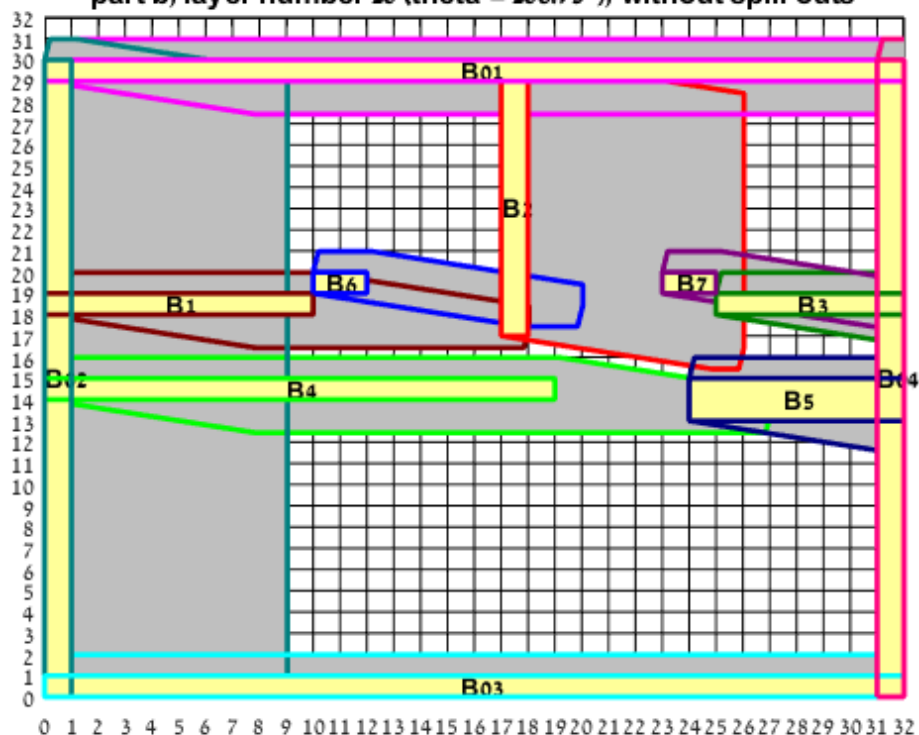
part b, layer number 8 ( $\theta = 78.75^\circ$ ), without spill outs



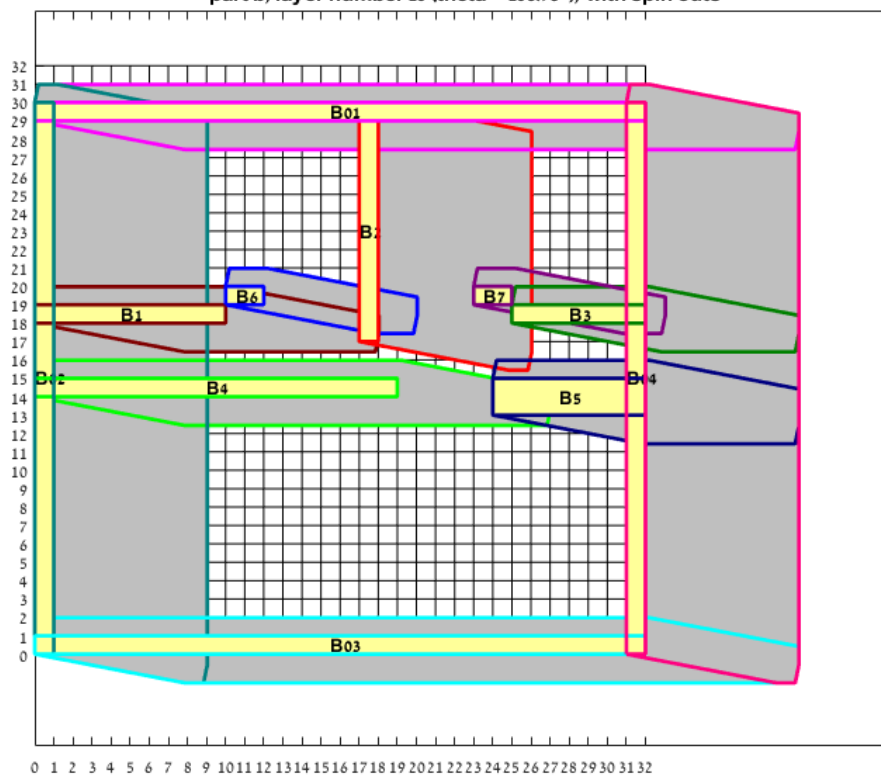
part b, layer number 8 ( $\theta = 78.75^\circ$ ), with spill outs



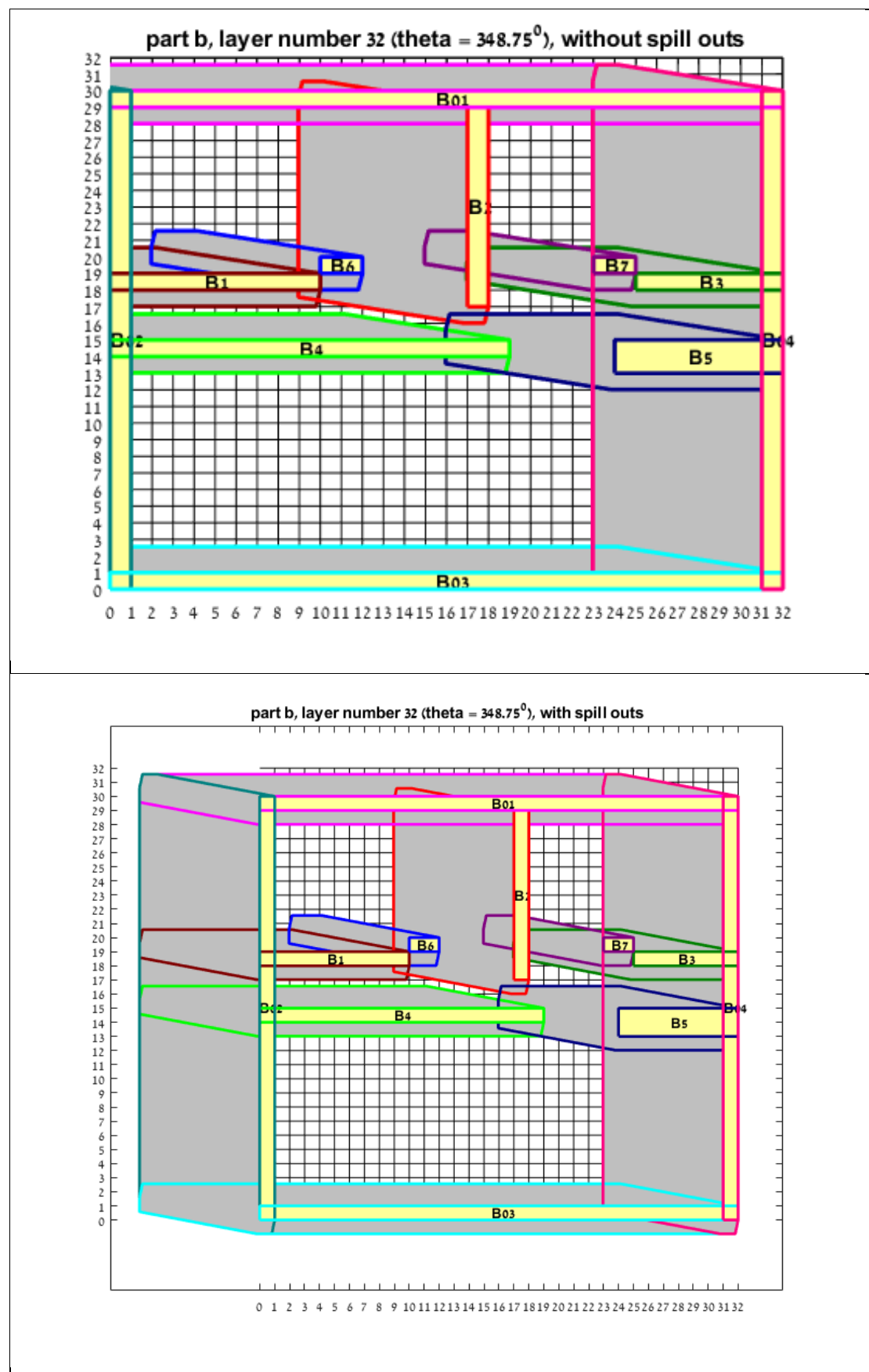
part b, layer number 16 (theta = 168.75°), without spill outs



part b, layer number 16 (theta = 168.75°), with spill outs







ניתן לראות שבכל השכבות התקבלה זליגה.

### שאלה 3

3. As a preparation for part #2 of the mini-project, set up an  $(x, y, \theta)$  grid of dimension  $32 \times 32 \times 32$ . For every  $\theta$  layer, fill the cells of the respective c-obstacles boundaries by 1. The others should be filled with 0. Note that you will need a code that draws a discretized straight line between two given endpoints. Note, too, that portions of the c-obstacles corresponding to the outer wall spill out of the grid. Give us a *printout of layers 1, 8, 16, 32*.

בשלב זה כתבנו פונקציה (OccupancyGrid) אשר יוצרת מפה דיסקרטית עבור כל חתך  $\theta$ :  
פסודו-קוד:

- לכל תא במפה המוגדר ע"י  $\{i, j\}$
- לכל מכשול  $k$
- **בדיקה** האם תא  $\{i, j\}$  מתנגש (**חודר**) למכשול  $k$ . הבדיקה מבוצעת באמצעות פונקצית מטלאב קיימת `intersect`.
- אם כן- סמן במפה 1.

להלן הפלט המתקבל עבור הרצת שכבות  $[1, 8, 16, 32]$  סביב מכשול כלל המכשולים. כל תא אשר חלק ממנו שייך למרחב הקונפיגורציה מסומן בערך 1 וצבוע **בטורקיז**. שאר המשבצות מסומנות ב-0 וצבועות בלבן. כל ציור מתאר מטריצה בגודל  $32 \times 32$ . **באדום** מסומנים המכשולים.

