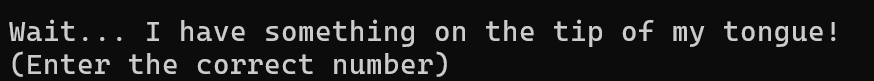
**Write up – Wonderland 4,5**

ת"ז: 216175398

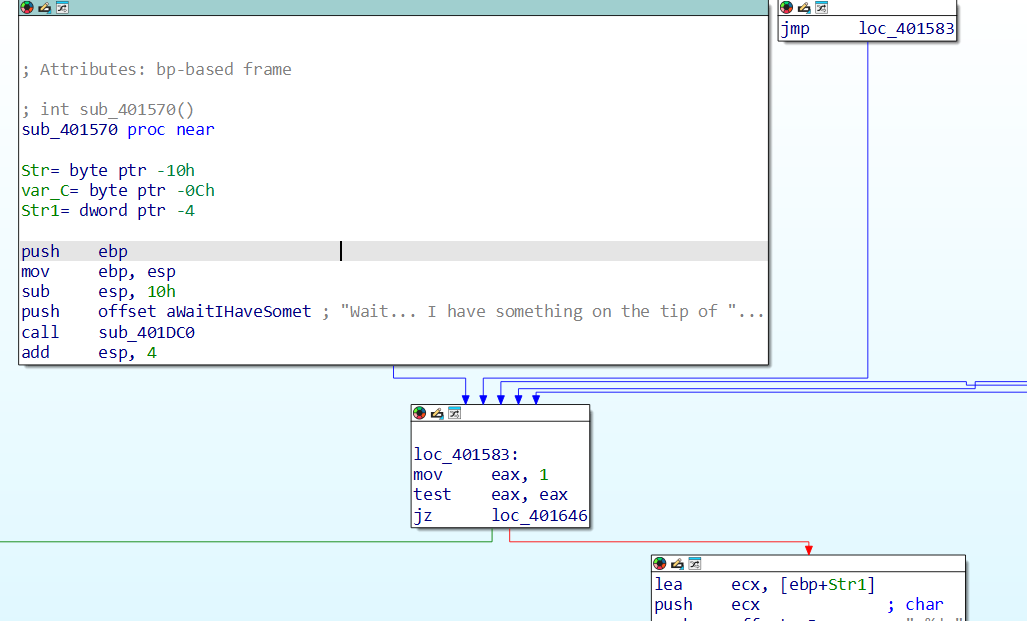
שם: יוסף חפץ

**שלב 4**

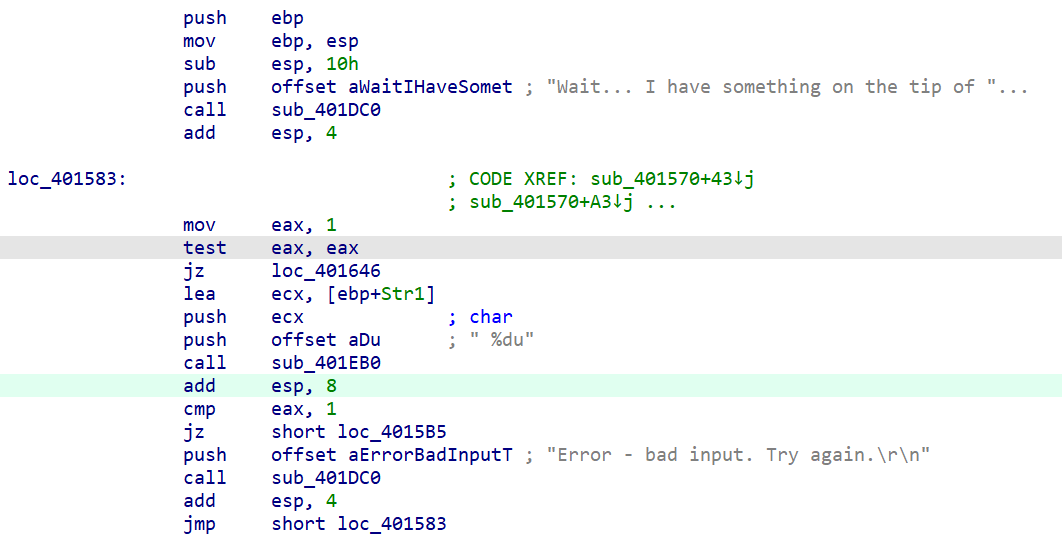
עכשיו אנחנו כאן:



ונקודת הפתיחה של התוכנית שלנו היא –



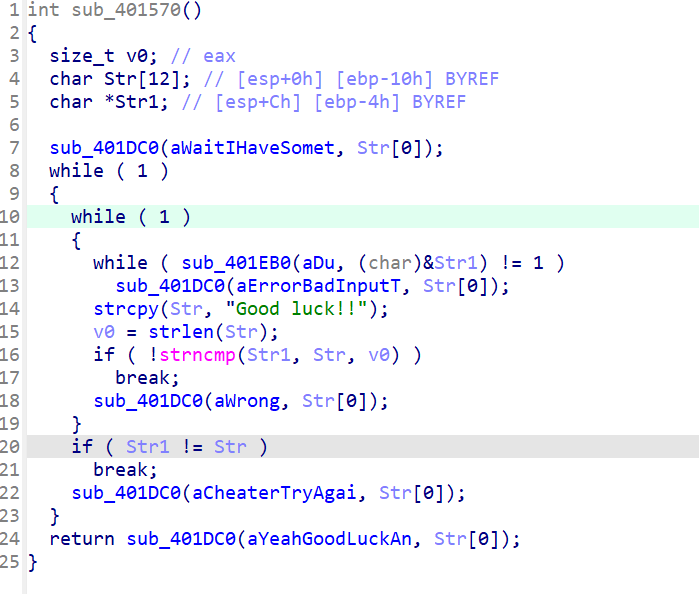
עיון קצר בקוד ייקח אותנו לתוכנית הבאה -



מה שקורה פה בעצם זה קלט מהמשתמש, ואם הוא קלט תקין קופצים ל - loc\_4015B5, אחרת נותנים הודעת שגיאה ומחכים לקלט הבא. לכן נעבור לכתובת הזאת וננתח אותה.

**loc\_4015B5**

עכשיו הגענו לחלק החשוב הבא [זה הקוד C שלו ב-IDA]:



החלק הזה הוא מה שבודק את נכונות הקלט, הוא מבצע 2 בדיקות:

1. האם המחרוזת שבמיקום שווה ל-"Good luck!!"
2. האם המיקום שונה מהמיקום שב-Str

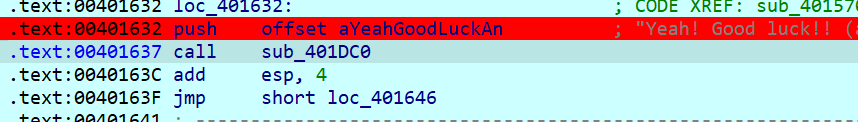
כדי שהקלט יעבוד צריך ששתי הבדיקות האלה יחזירו אמת. לכן בעצם הקלט צריך להיות כתובת במחסנית (Stack) שמכיל את הערך "Good luck!!" ושונה מהכתובת Str.

כעת נמצא בקוד את המיקומים של המחרוזות שלנו –

**תמונה שמכילה טקסט, גופן, צילום מסך

תוכן בינה מלאכותית גנרטיבית עשוי להיות שגוי.**

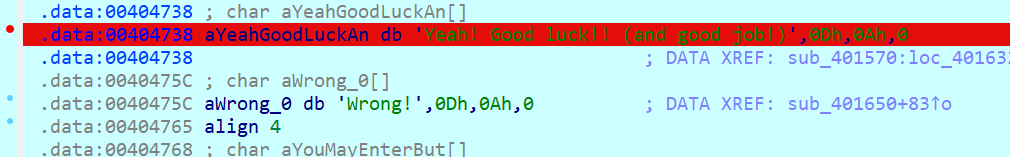
המיקום של הראשון הוא -

****

עכשיו נדבג ונראה את המיקום שלו במחסנית – 1703676.

נכניס את הערך הזה ונקבל - 

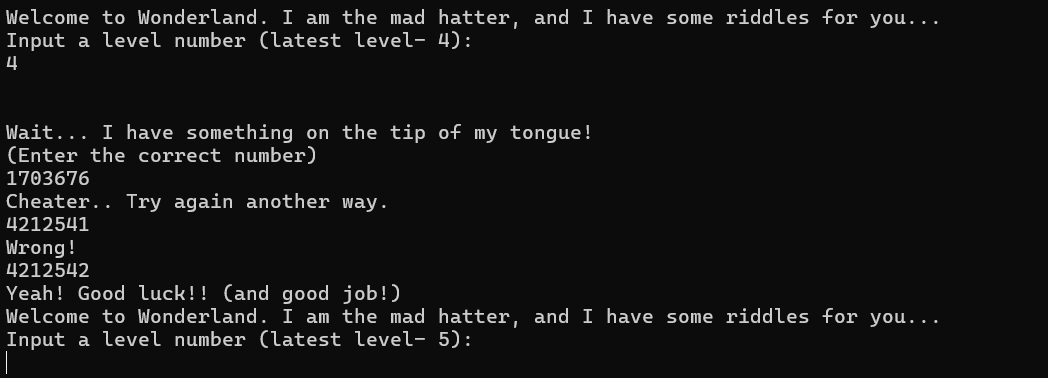
לכן ננסה את הערך השני –



המיקום שלו במחסנית הוא – 4212542 נכניס אותו **וכן זה עובד!!**

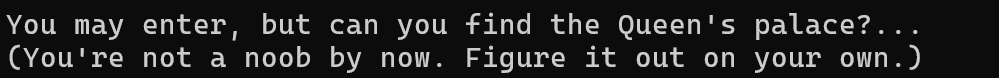
**סיכום**

הסיסמה היא - 4212542  
הקלט הזה מצביע על "Good luck!!" שב-data section, ולכן עובר גם את בדיקת התוכן **וגם** את בדיקת האנטי־צ'יט.

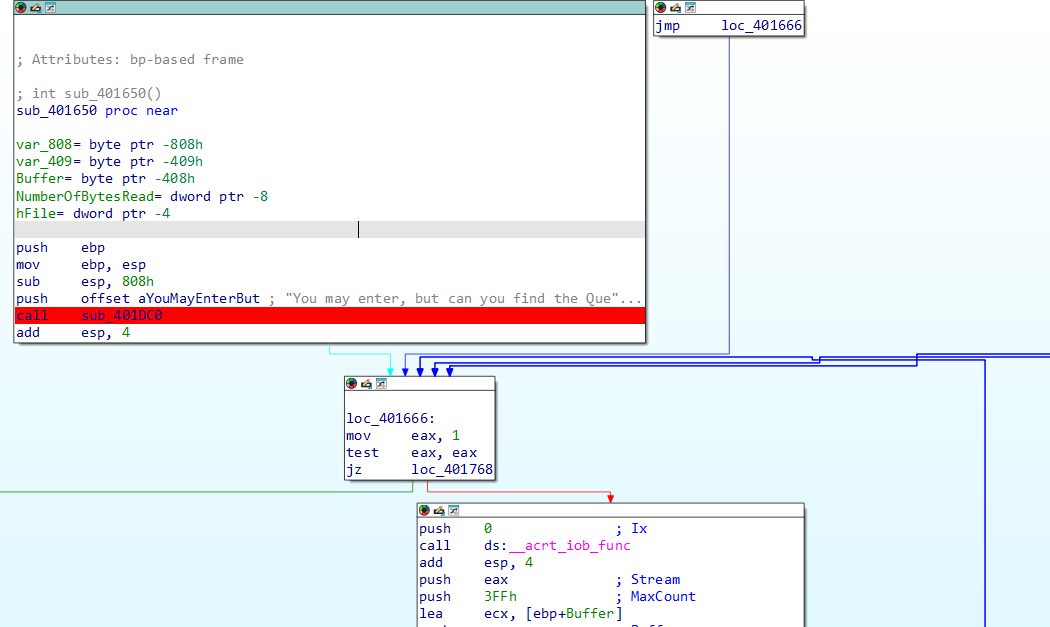


**שלב 5**

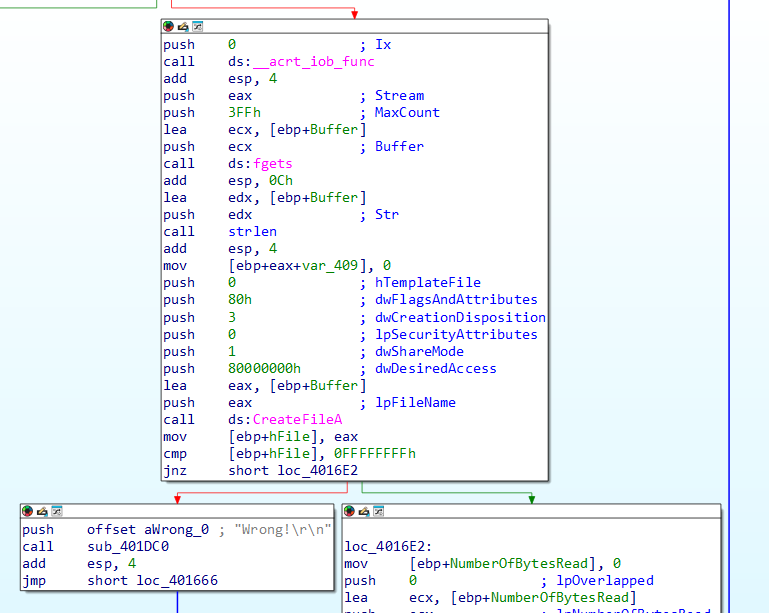
עכשיו אנחנו כאן:



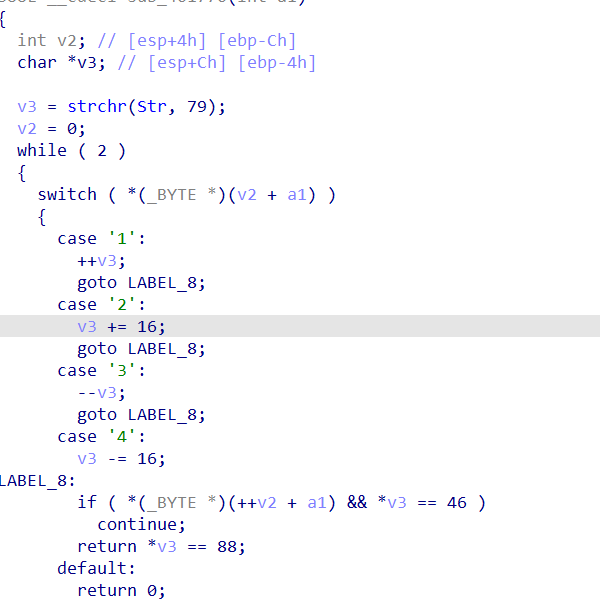
והקוד שלנו נראה כך:



ישר רואים שיש כאן קשר לקבצים, התוכנית מקבלת כתובת של קובץ ובודקת אותו –



אחרי שבודקים תקינות קוראים ל **sub\_401770** על התוכן של הקובץ, הנה הקוד C שלו:



**ניתוח הפונקציה sub\_401770**

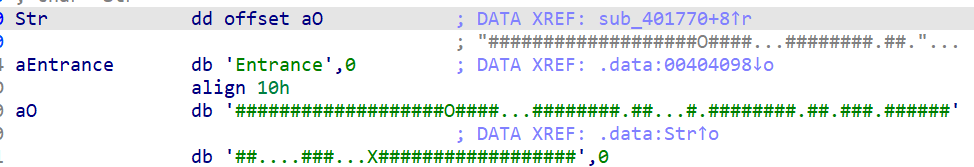
1. הפונקציה מקבלת מצביע לbuffer-.
2. היא מבצעת מעבר על כל תו בקלט ומזהה תווים בין 1 ל-4.
3. עבור כל תו, היא מבצעת פעולה בהתאם:
   1. '1' : מוסיף 1 למצביע
   2. : '2' מוסיף 0x10 למצביע
   3. '3' : מקטין ב-1 את המצביע
   4. : '4' מקטין ב-0x10 את המצביע
4. הפעולות מתבצעות על המשתנה הגלובלי Str, שמצביע על מפת מבוך (כמו מערך דו-ממדי).

**מה קורה בפועל?**

* בכל שלב, הפונקציה קוראת את התו הבא ב־ Str אחרי השינויים.
* אם היא מוצאת נקודה '.' – ממשיכה בלולאה.
* אם היא מוצאת את התו 'X' – זה סימן שסיימנו את המסלול בהצלחה, והפונקציה מחזירה 1.
* אם לא – היא מחזירה 0.

עשיו ניצור את הקובץ טקסט שמכיל את רשימת המספרים שייתנו את המסלול הרצוי, אבל איך מוצאים אותו?

רואים שיש פה איזשהו משתנה Str –



המשתנה הזה הוא המפה שלנו! עשיו צריך רק למצוא את הניווט הנכון, בשביל זה בניתי אתר נחמד:

**2221114411411222111**

שזו בעצם רשימת הפקודות שמנחות את הפונקציה לנווט דרך המפה בהצלחה.

**סיכום:**

על-ידי ניתוח הפונקציה sub\_401770, הבנו שהקלט הרצוי הוא מחרוזת של תווים '1' עד '4' שמייצגים תנועות בתוך מפת מבוך. רק אם נבצע את הרצף הנכון נגיע לתו 'X' וכל סטייה תוביל להודעת שגיאה. לאחר בחינה של הפונקציה והנתונים במשתנה 'X' מצאנו שהקלט הנכון הוא:

**2221114411411222111**

בשביל לעזור לעצמי בניתי אתר שיעזור להמחיש את השביל - <https://wonderland-solver.onrender.com/>

וזה הפתרון! 🧠

