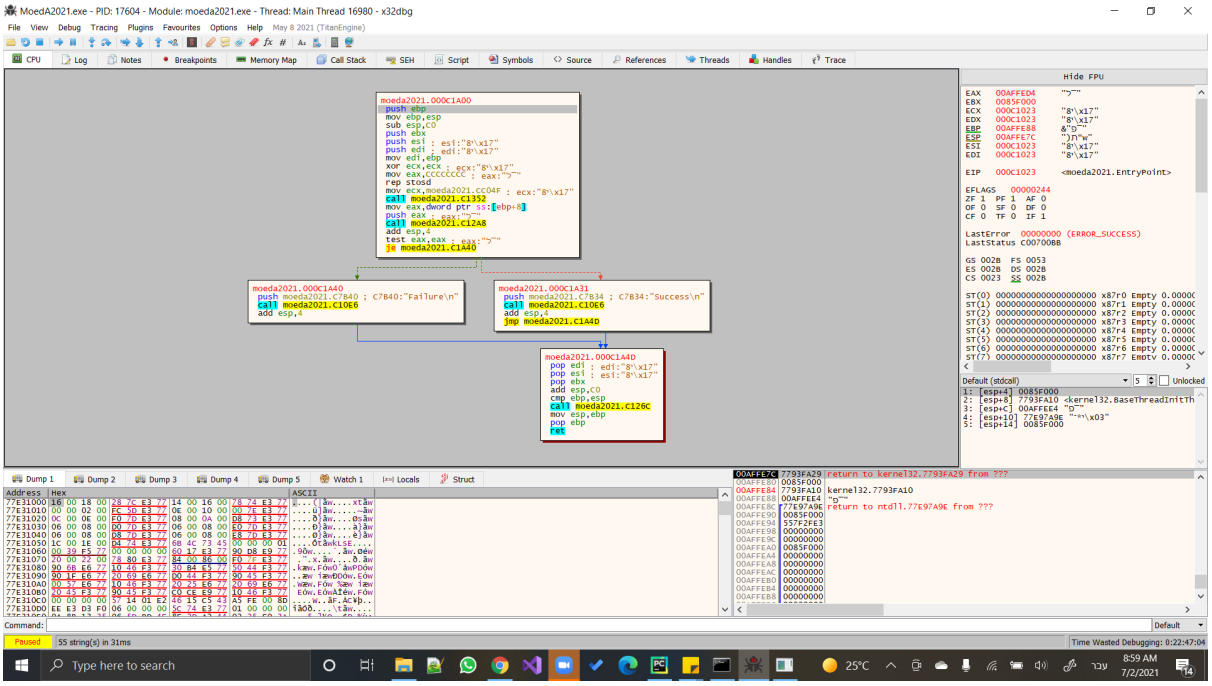
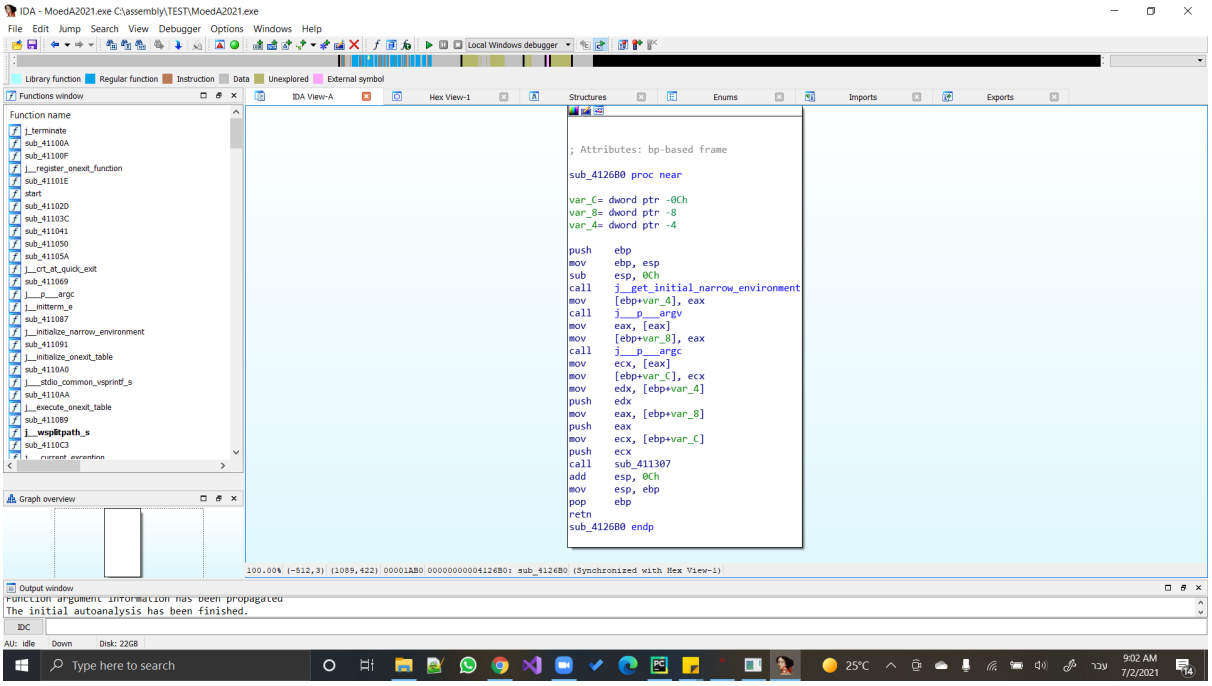
## מבחן ברברסינג

הרצתי את התוכנה בCMD , הכנסתי קלט אקראי והודפס לי FAILURE, פתחתי את הקובץ בX32DGB
וראיתי שיש שתי מחרוזות : FAILURE ו SUCCESS, אלו הן המחרוזות החשובות.
מצאתי בעקבות כך את הפונקציה הזו:

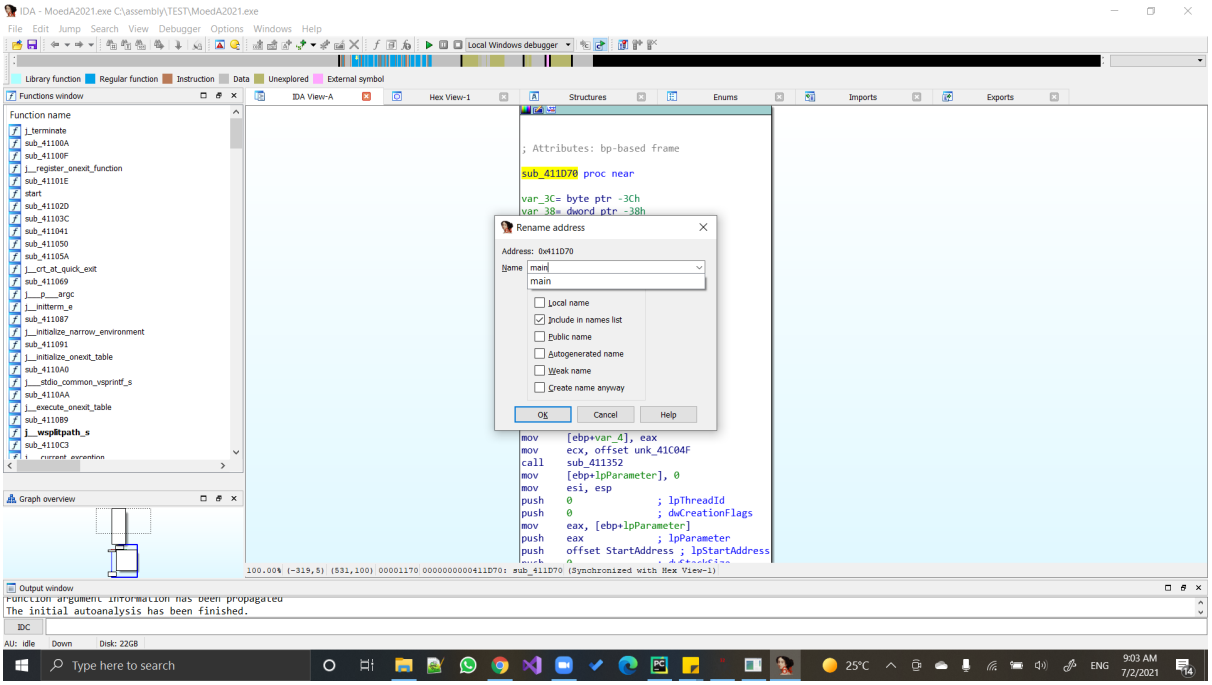



ניסיתי מכאן לגלות את מנגנון האנטי-דיבאג, זה לא היה לי ברור במיוחד ולכן העדפתי בשלב זה לעבור לניתוח סטטי בIDA :

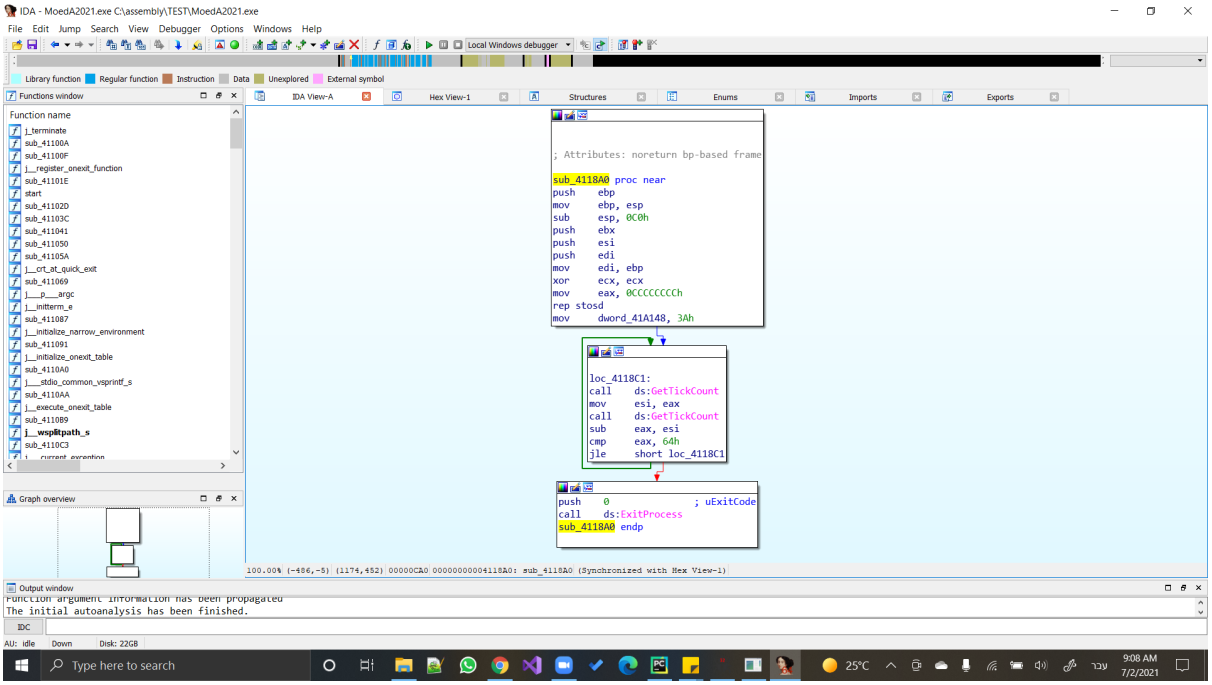
פתחתי את הקוד בIDA כדי למצוא את הMAIN, הלכתי תמיד הכי קרוב לEXIT ובסוף מצאתי את הדפוס הבא:



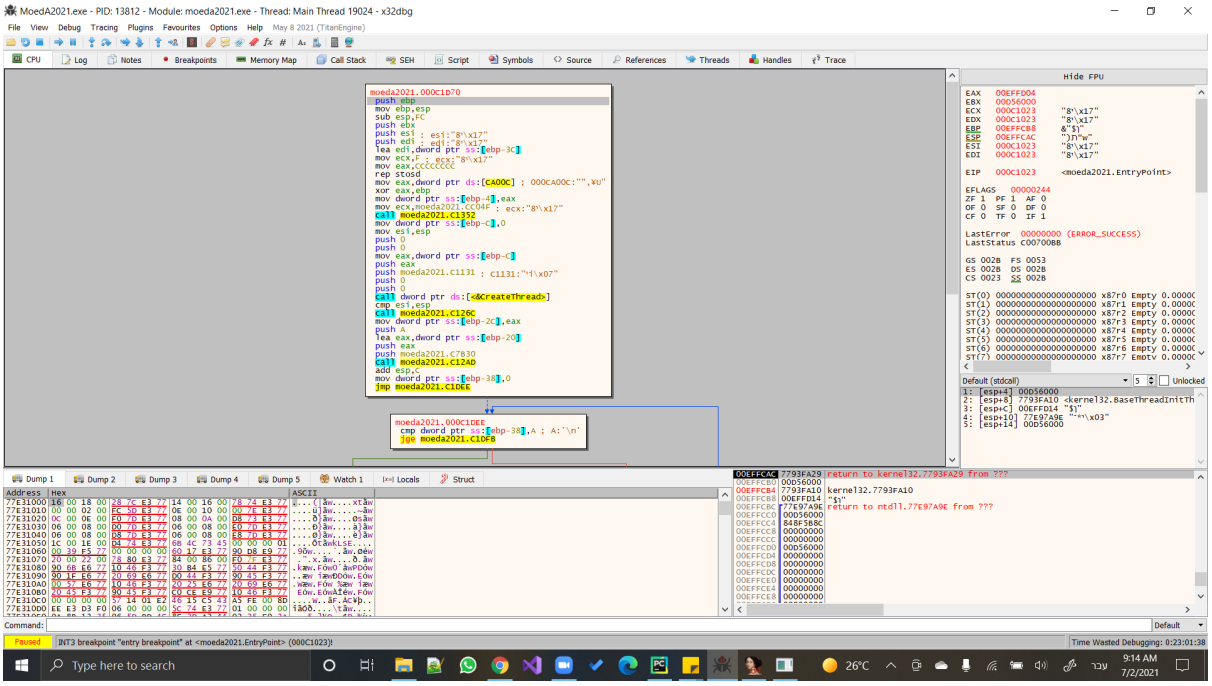

שנקרא לפני הMAIN, לכן הMAIN נמצא בכתובת 411D70



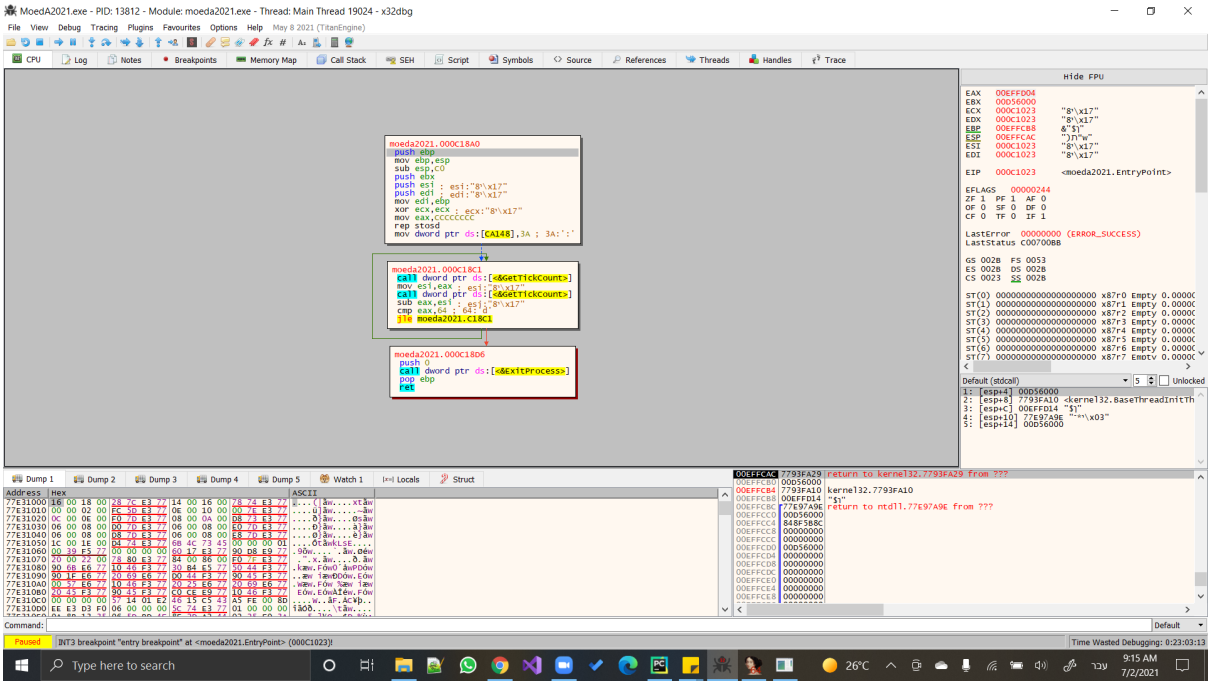

האם אני חולמת או שמא בתוך הMAIN יש קריאה לפונקציה CREATE THREAD ?
בהחלט יש קריאה כזו, כיוון שביצירת THREAD צריך לשלוח לו את הפונקציה שהוא צריך לבצע, אעיין בפרמטרים שנדחפו למחסנית לפני הקריאה לפונקציה שיוצרת את הTHREAD, אחד הפרמטרים הוא פונקציה שכתובתה (כלומר הכתובת של הפונקציה הסופית, לא של זו שמתווכת ע"י קפיצה) 4118A0 :



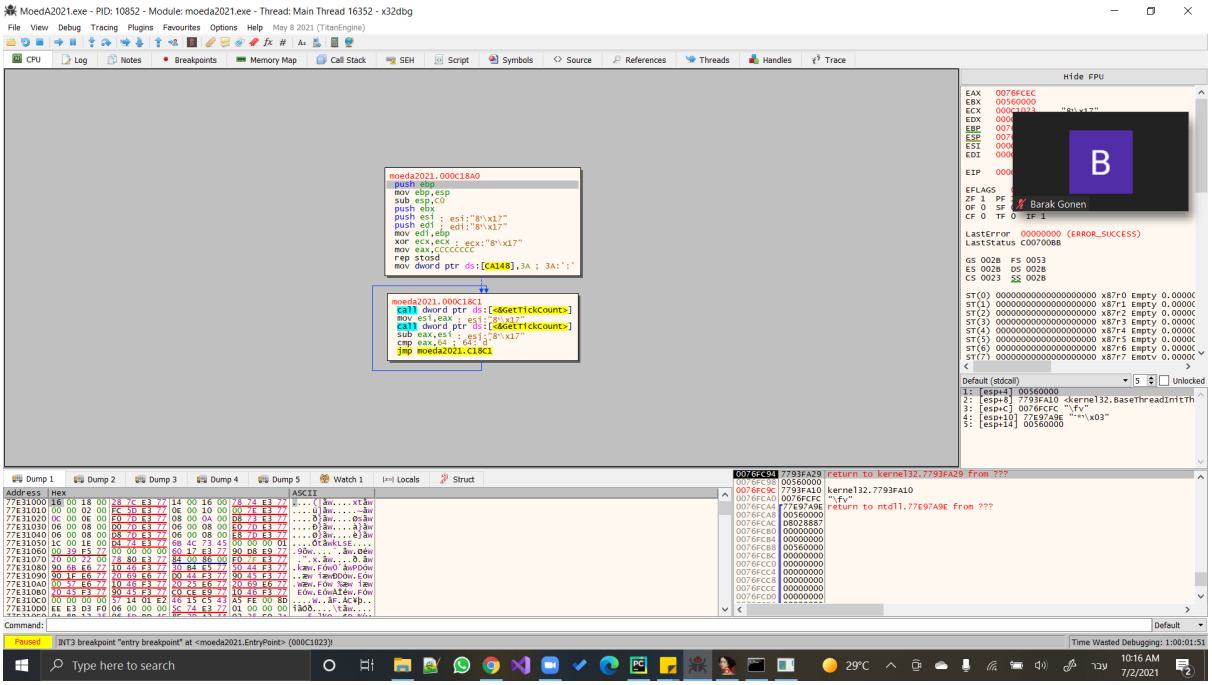

ניתן לראות שפונקצית האנטי דיבאג משתמשת בטכניקה של מדידת זמנים. הפונקציה מבצעת שתי דגימות של זמן ע"י GETTICKCOUNT, ובודקת את ההפרש ביניהן מתוך הנחה שאם אנחנו נמצאים במהלך תהליך דיבוג, סביר ויעבור פרק זמן ארוך בין שתי פקודות שלא דורשות כ"כ הרבה זמן. אם עבר הרבה זמן, הפונקציה EXITPROCESS תיקרא ונצא מהתוכנית.




תחילה מצאתי את הMAIN בX32DBG (כיוון שזה ניתוח דינאמי אז פה הכתובת היא C1D70).
והנה הפונקציה שמודדת זמנים:




בהתחלה חשבתי לשים פה RET, וזה באמת מה שעשיתי בהתחלה, אבל בהמשך גיליתי משהו על כך שיש שימוש במשתנה DWORD41A148, והפונקציה שלנו מאתחלת אותו כך שהוא מכיל את הערך 3AH, לכן אי אפשר לעשות פשוט RET וצריך לפצפץ בצורה מתוחכמת יותר.
את הבדיקה שיש שם - jle, אני פשוט משנה לJMP רגיל. ככה הוא ימדוד זמן עד הנצח ואף פעם לא יבדוק את ההפרש ביניהם, ומצד שני האתחול של 41A148 יעשה. כך:




מקסים ומלבב!

כעת ננסה להבין את אופן פעולת התכנית:

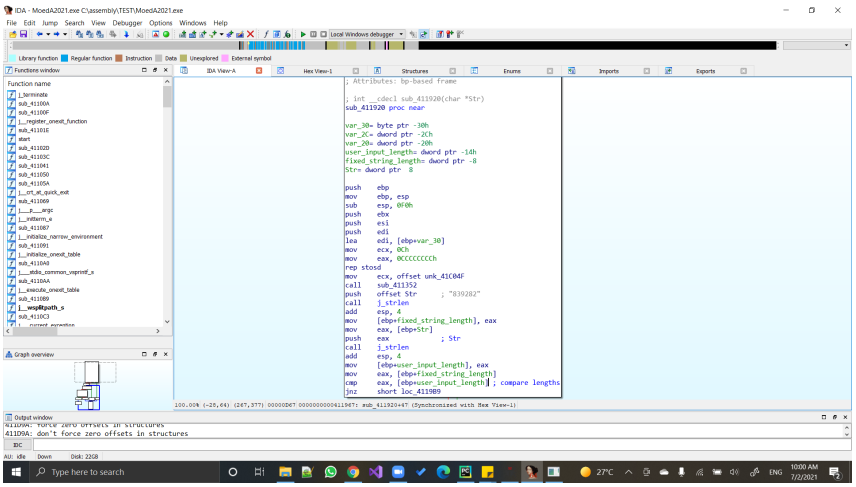
אחרי כל ההאתחולים ויצירת הTHREAD המרושע, אני רואה שיש קריאה לפונקציה שקולטת קלט מהמשתמש ( כשאני מדבגת אני רואה שהוא לא נותן לי לעשות STEP OVER עד שאני מזינה קלט בCMD ובנוסף כשנכנסתי אליה ראיתי acrt iob func). ניתן לראות שיש לנו לולאה -

אני רואה לולאה עם COUNTER שרצה 10 פעמים, הלולאה הזו קוראת לפונקציה שמדפיסה קו נטוי, ממתינה קצת ומדפיסה קו נטוי הפוך, זה החלק הגרפי הפחות מעניין ולכן ארצה לבדוק מה קורה בתום הלולאה הזו:

אחרי הלולאה יש קריאה לפונקציה שמצאתי בהתחלה ( זאת שמתפצלת לSUCCESS או FAILURE). שהפונקציה מקבלת את הקלט של המשתמש. הפונקציה הזו קוראת לפונקציה נוספת, ניכנס אליה:

ניתן לראות שתי קריאות לSRTLEN :

הקריאה הראשונה היא עם המחרוזת "839282" והשניה עם הקלט של המשתמש, לאחר מכן שני האורכים מושווים זה לזה:



במקרה שאורך המחרוזות זהה, יש לנו לולאה שVAR_2C הוא המונה שלה והוא בעצם רץ מ0 עד 6 (כאורך המחרוזת הקבועה) וככה ניגשים תו אחר תו למחרוזת שהכניס המשתמש.
אני שמה לב שבהמשך ניגשים למקום בזיכרון שנקרא DWORD41A148, כשאני עושה לו XREF אני רואה שפונקציית האנטי דיבאג מכניסה בו ערך של 3AH (לכן שיניתי את הפצפוץ).
עבור כל תו מוצאים את ההפרש בין האסקי שלו לבין 3A, ואחר כך לוקחים כל תו בfixed string ומחסירים ממנו 30 ואז משווים בין שתי התוצאות.
כלומר עבור כל תו i בסיסמא צריך להתקיים:

0x3A - Passowrd[i] = fixed_string[i]-0x30

אחרי שגיליתי את זה עשיתי קצת חישובים עם עט ועיפרון וגיליתי שהסיסמא צריכה להיות:
271828
וראו איזה פלא -

```
C:\assembly\TEST>MoedA2021.exe
271828
Success
```

הסוף.

נ"ב-
תודה על קורס נחמד בהחלט :)