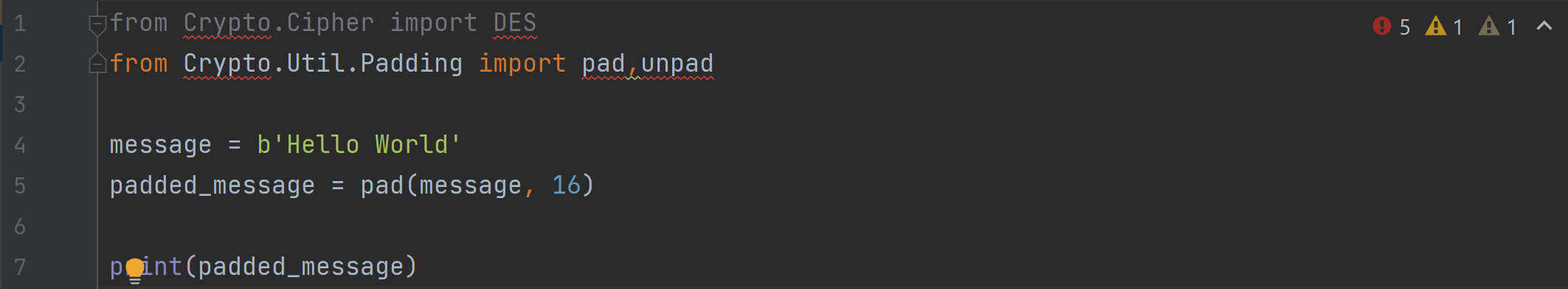
**הסבר על הקוד**

**שלב 2:**

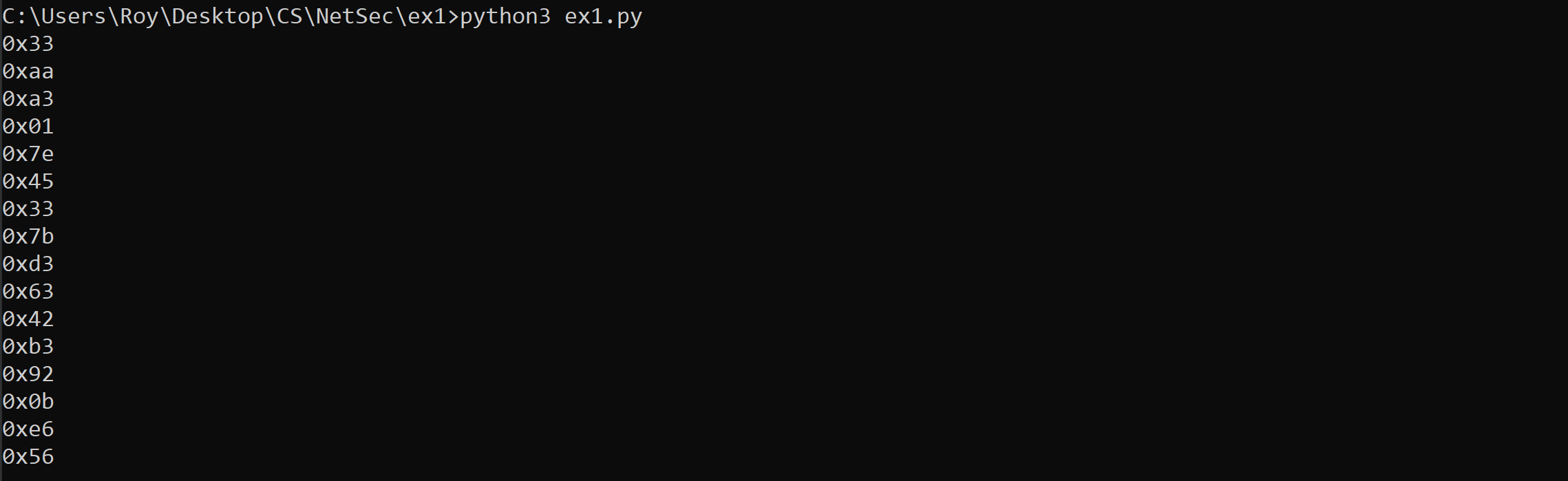
ניצור את ההודעה ונרפד אותה בעזרת הפונקציה pad ואכן:  


**שלב 3:**

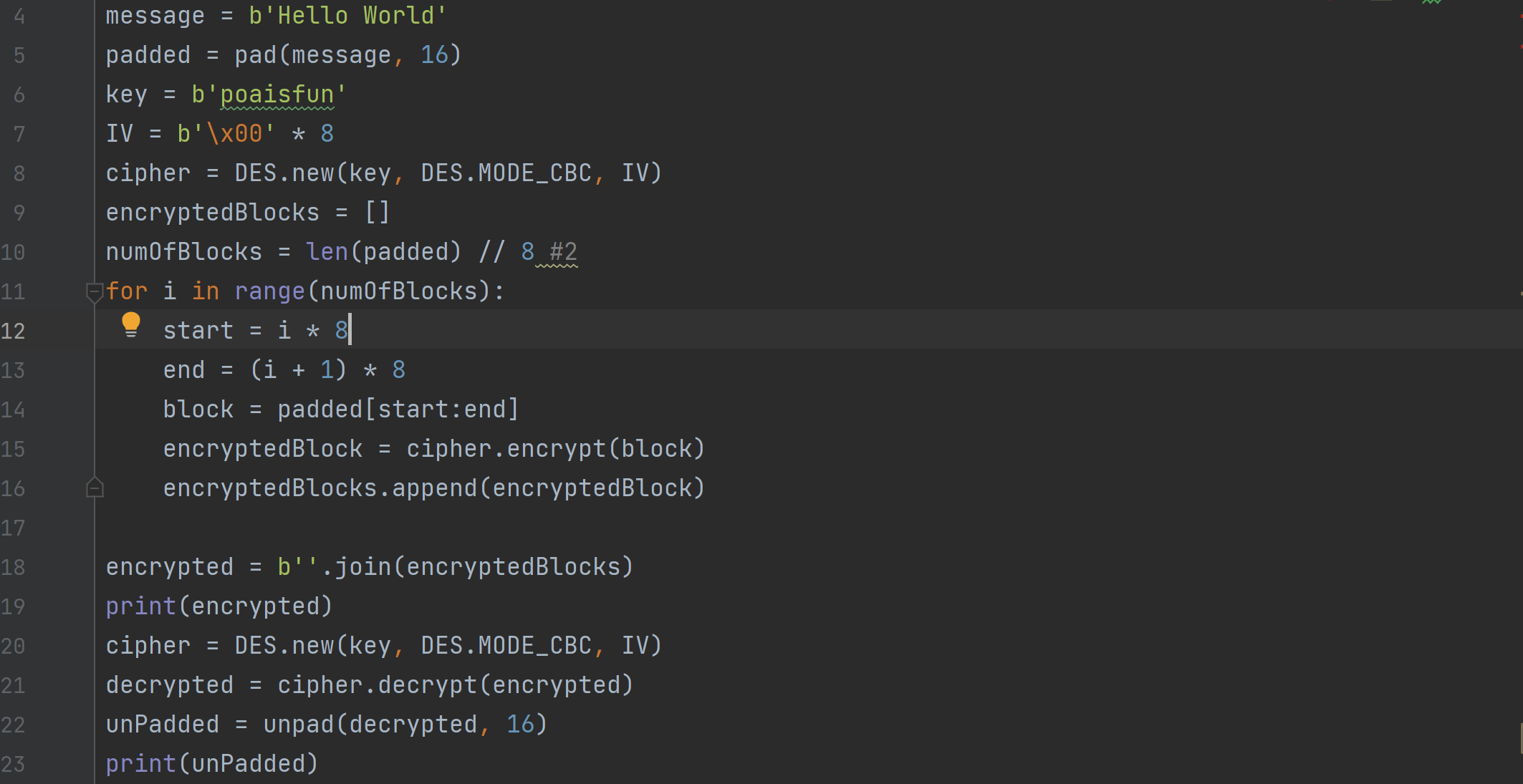
****

כעת נגדיר את המפתח poaisfun כמבוקש ואת IV להיות 00\*8 כי DES בבלוקים בגודל שמונה ונבצע חלוקה לבלוקים לפי כמות בלוקים של אורך ההודעה חלקי 8 (בחילוק שלמים), ניצור את הcipher שלנו בשורה 8 ונבצע הצפנה בלולאה על הבלוקים בשורות 11-16 ונוסיף למערך של הבלוקים המוצפנים שלנו, לבסוף נכניס הכל להודעה ארוכה ב18, נבצע המרה להקסאדצימלי ונדפיס כל פעם שני תווים כי בכל בית יש שני תווים הקסאדצימליים, נדפיס עם הקידומת 0x לקריאה נוחה יותר.

ואכן-

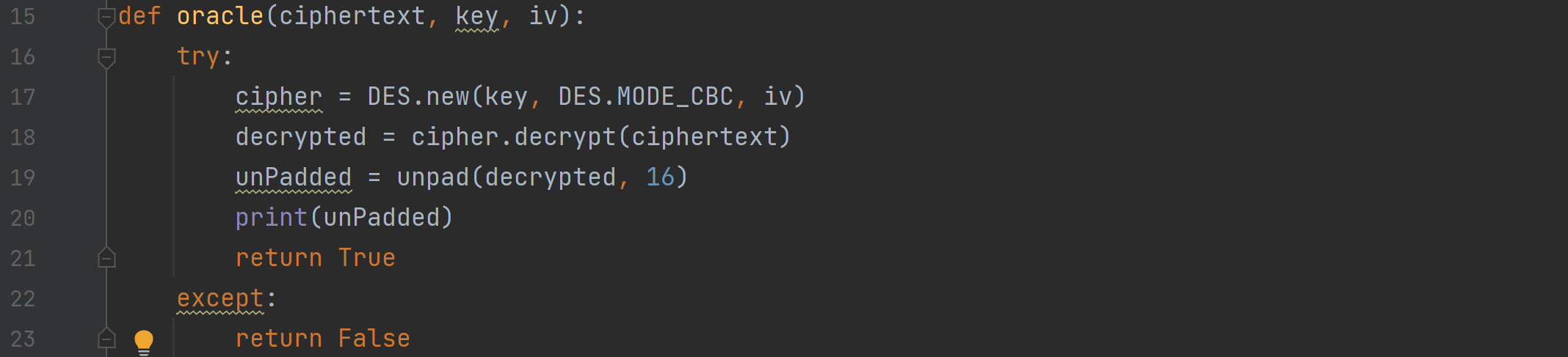
****

ההצפנה עבדה והבתים כמו שהיו אמורים לצאת.

**שלב 4:  
**נבצע הצפנה כמו קודם, כעת ניצור cipher חדש לפיענוח כי אי אפשר באותו אחד, ונפענח, לבסוף נבטל את הריפוד ונדפיס לפני הפיענוח ואחרי ונקבל אכן:  

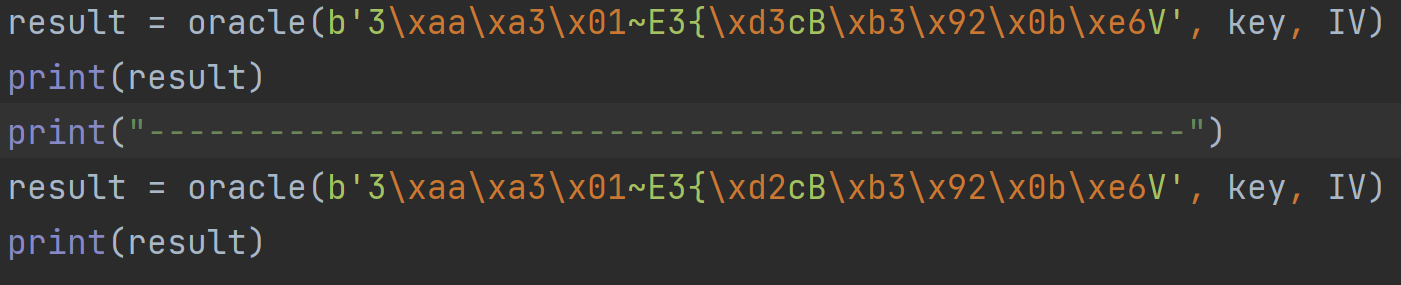

**שלב 5:  
**

על ידי פעולת xor המוגדרת, בbitwise ואכן ההדפסות כנדרש:  

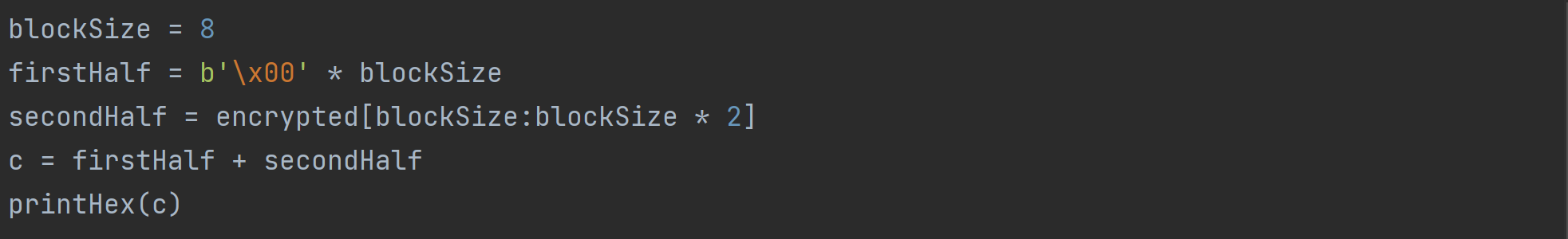

**שלב 6:  
**

באופן דומה לקודם, הפונקציה יוצרת cipher חדש עם הkey,iv ומבצעת פיענוח על הciphertext, במידה והפיענוח לא הצליח אז ניכשל, ואם זה ייכשל שם נגיע לexcept. נוסיף הדפסה בשביל שנוכל לראות אם זה תקין.

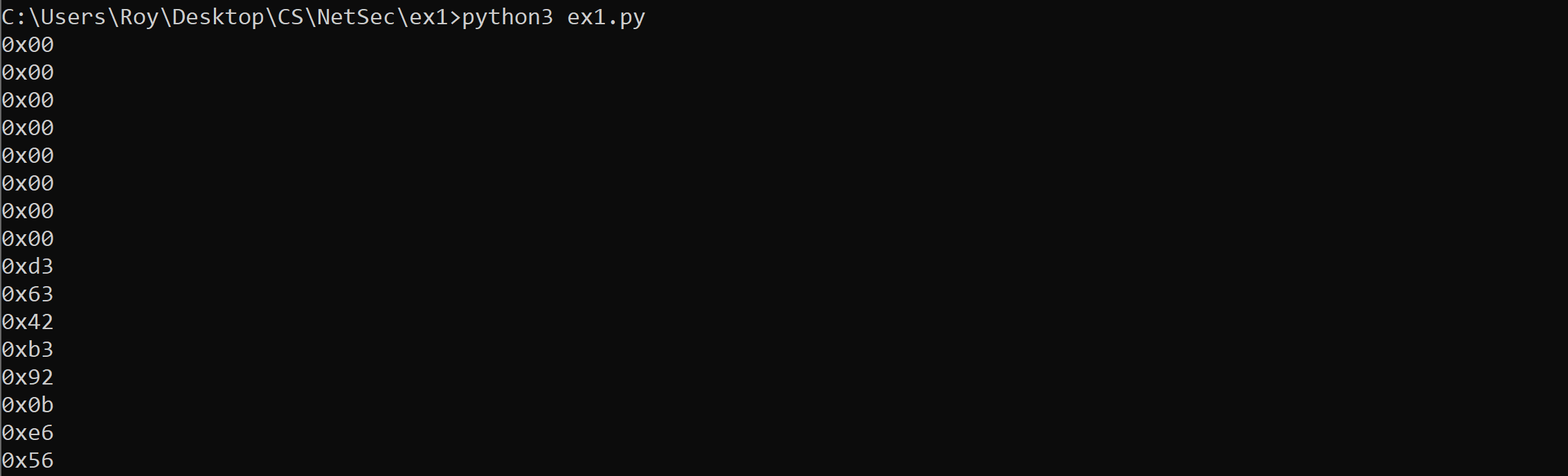
נבצע ניסיון:



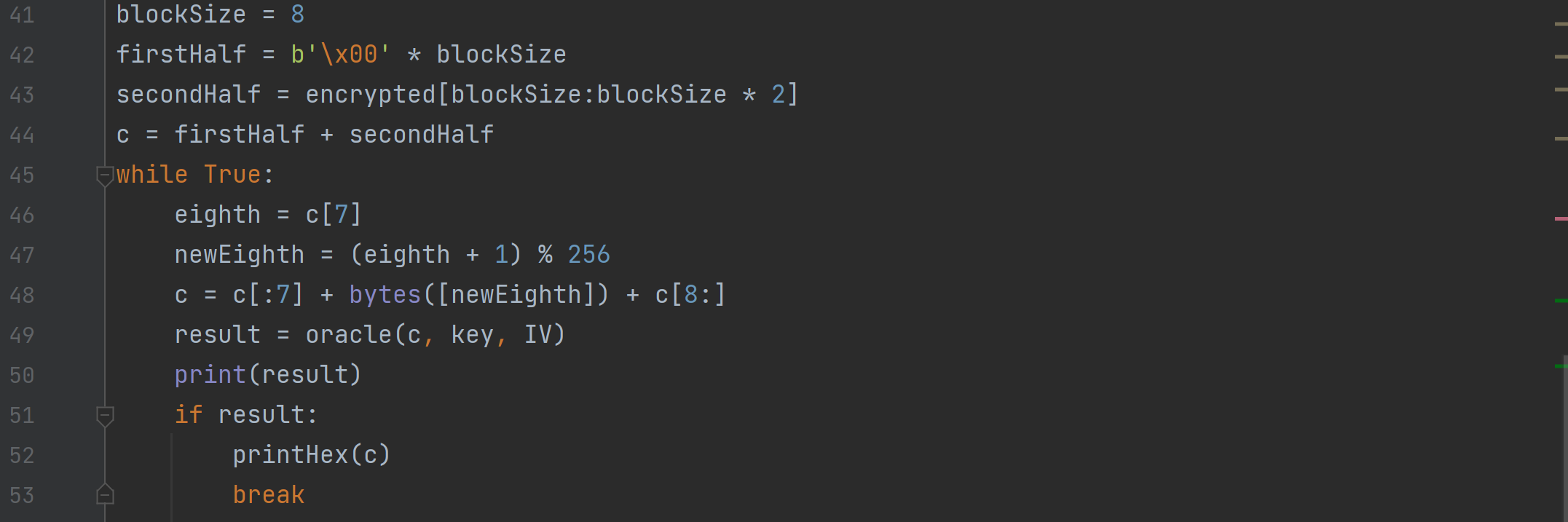
(הראשון זה עם הciphertext המוצפן מקודם, שהמקור זה Hello World ובשני שיניתי את ה\xd3 ל\xd2 ואמור לא לעבוד) ואכן:  

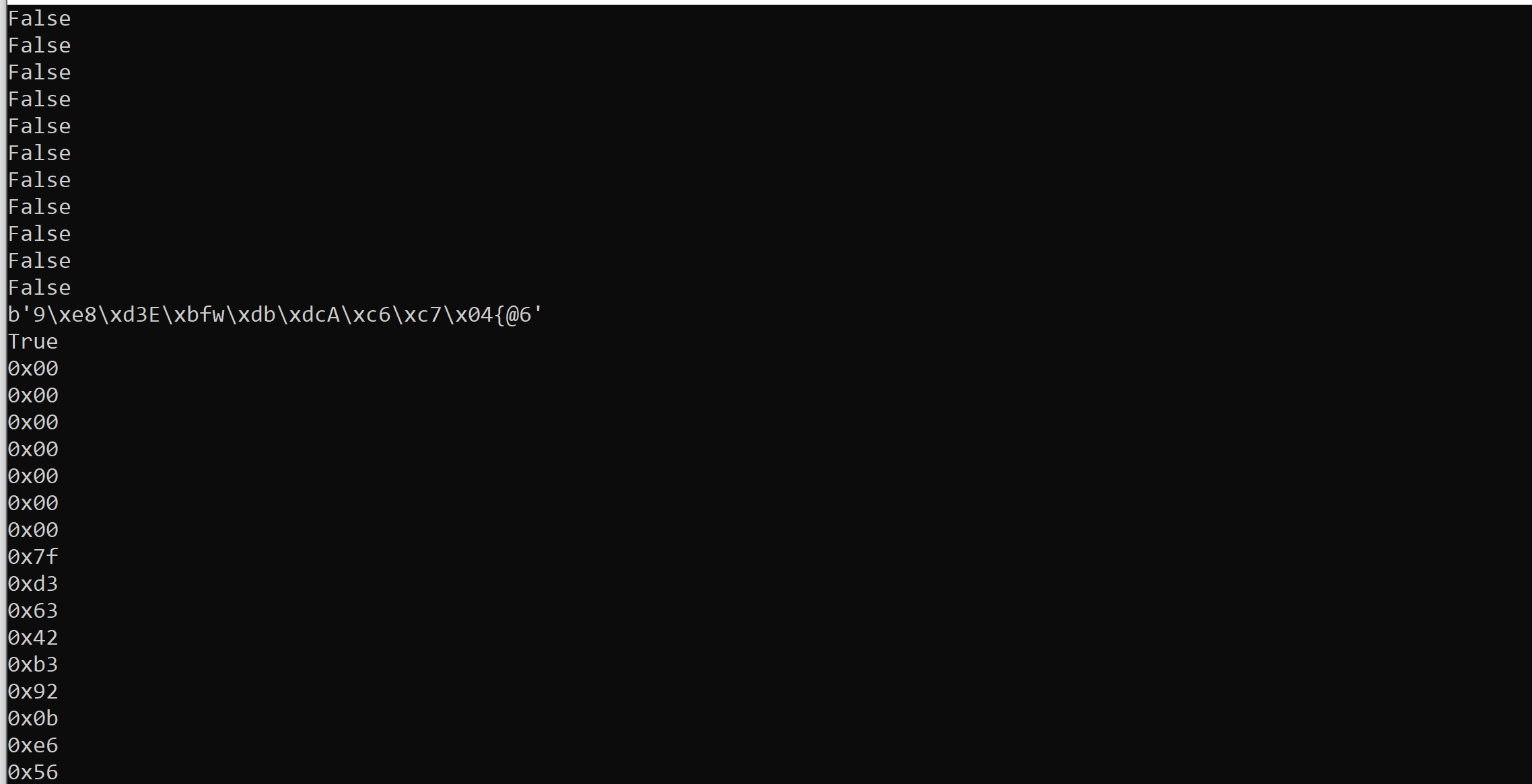

**שלב 7:**

השמות פשוטות כמבוקש, כאשר printHex זאת הפונקציה עם הלולאה המוזכרת בשלב 3 שמדפיסה בית-בית בהקסאדצימלי.

ואכן:  


כמבוקש.

**שלב 8:  
**

הוספתי כאן את הלולאה- בשורה 46 ניקח את הבית השמיני ונגדיל אותו בשורה לאחר מכן (יש שם מודולו 256 משום שיש רק 256 אופציות לבית 0-255), נעדכן את c בהתאם וננסה לפענח עד שהתוצאה תהיה נכונה, נדפיס את התוצאה בכל פעם לבקרה ולבסוף את המקרה שעבד:  


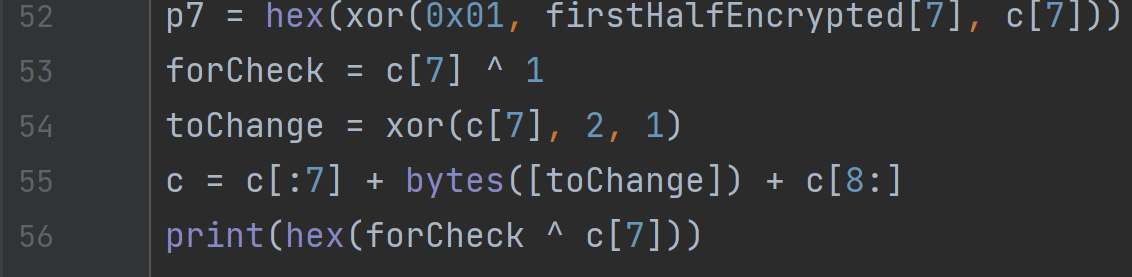
אחרי רצף של הרבה falseים נקבל את הtrue (ויש את ההדפסה שנשארה בoracle משלב 6.

**שלב 9:**המשוואה:  


לכן בשביל לקחת את נעשה xor על כל הגורמים:  

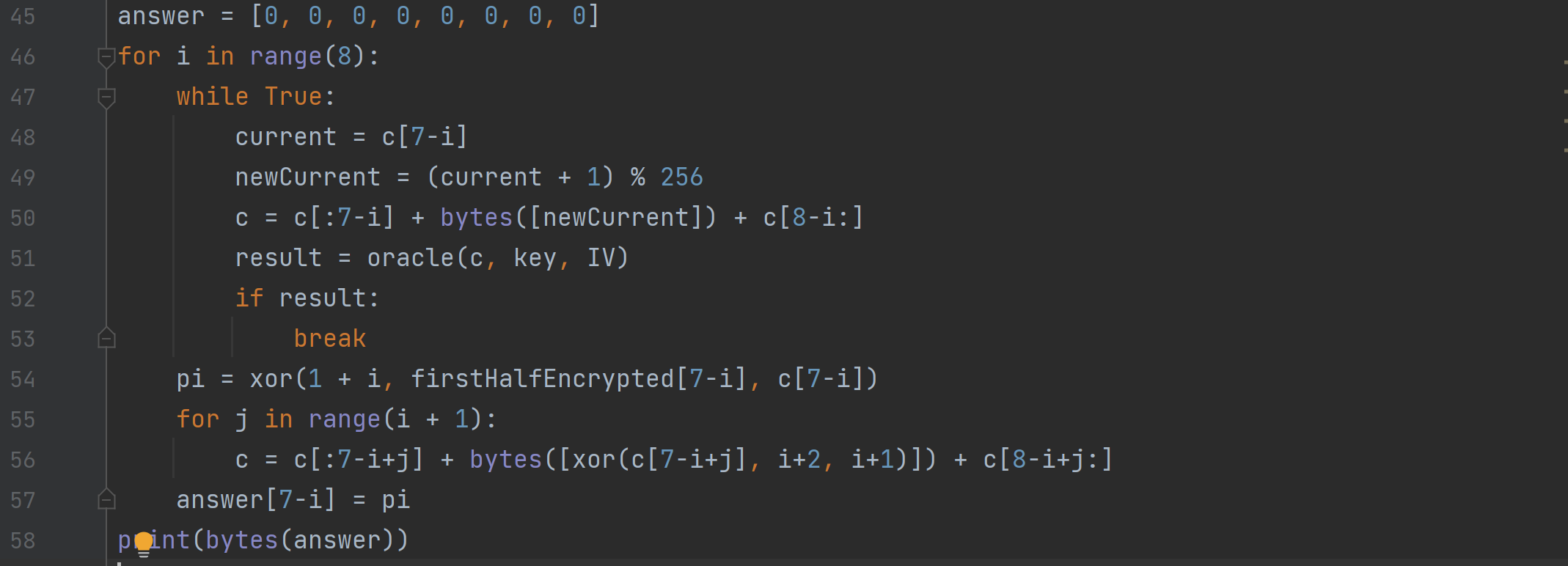

לאחר שבלולאה נמצא את התו שאיתו זה כן עובד, ניקח כל הגורמים בהתאם (הfirstHalfEncrypted כאן זה הבלוק הקודם).

ואכן:  
**

***שלב 10:***

*לאחר שp7 אכן פוענח כראוי, נשמור לבדיקה בהדפסה את c[7]^1 (בשביל לפענח עם xor על כל הגורמים, ונשנה את c[7] להיות , משום שעל פי הנוסחה מהמצגת, משום ש2 זה המבוקש שלנו ו1 זה הplaintext הקודם וככה נבטל אותו, ואכן בבדיקה עם נקבל כנדרש.*

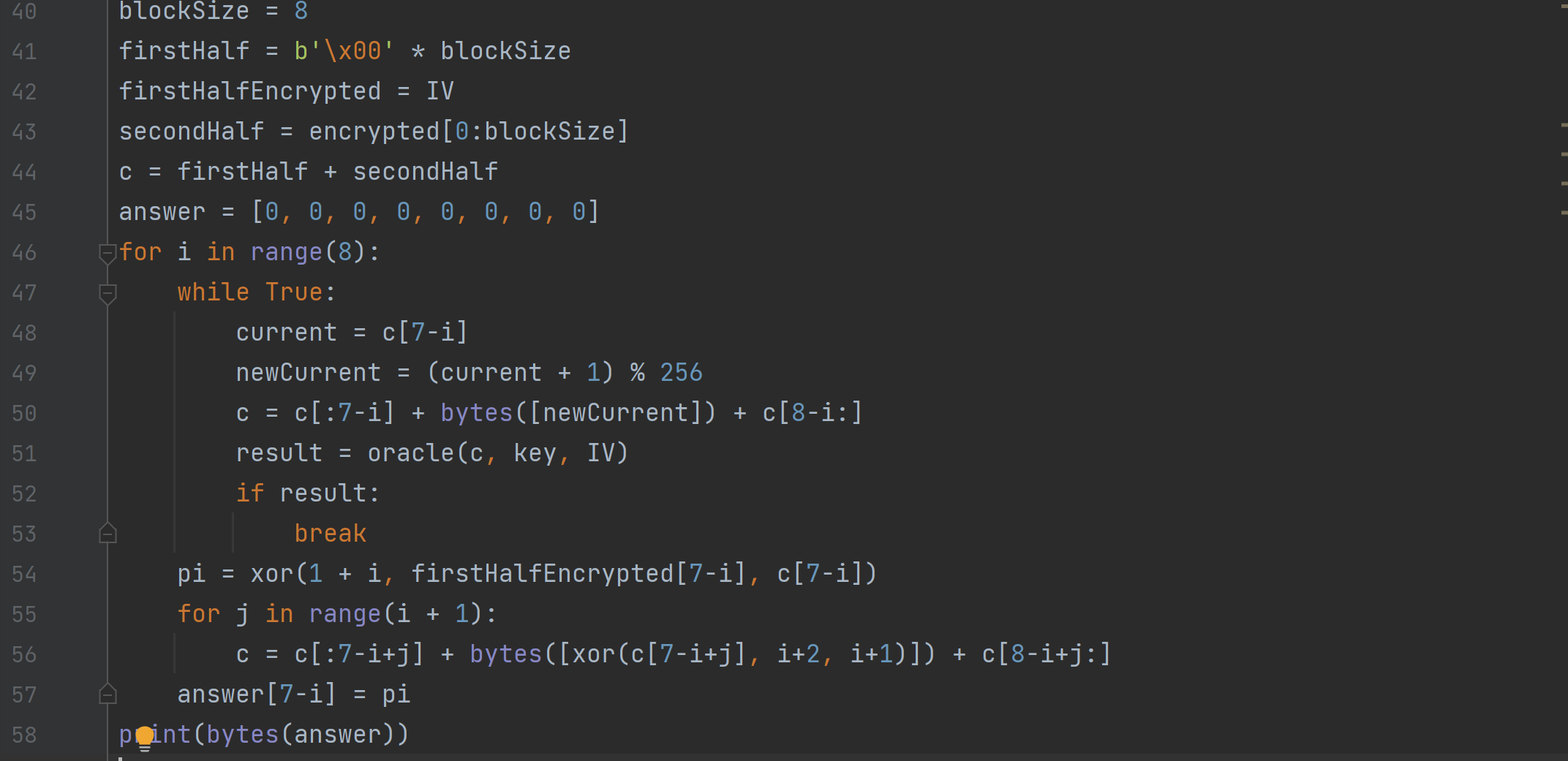
*ואכן:  
*

***שלב 11:***נכניס ללולאה ונהפוך הכל לכללי יותר עם תלות במיקום:  


בשורה 45 נאתחל מערך לשמירת התשובות, שורות 48-53 מוכרות לנו מקודם (רק השם eight השתנה לcurrent שכן עכשיו זה לא השמיני), ב54, הגורם הראשון בxor שונה לi+1 שכן אנחנו כל פעם משנים את הבית שיתפענח לplaintext אחר, וב55 בכל פעם נצטרך לשנות את כל הבתים האחרונים שיתאימו לאיטרציה הבאה ונשמור את התשובה שלנו.

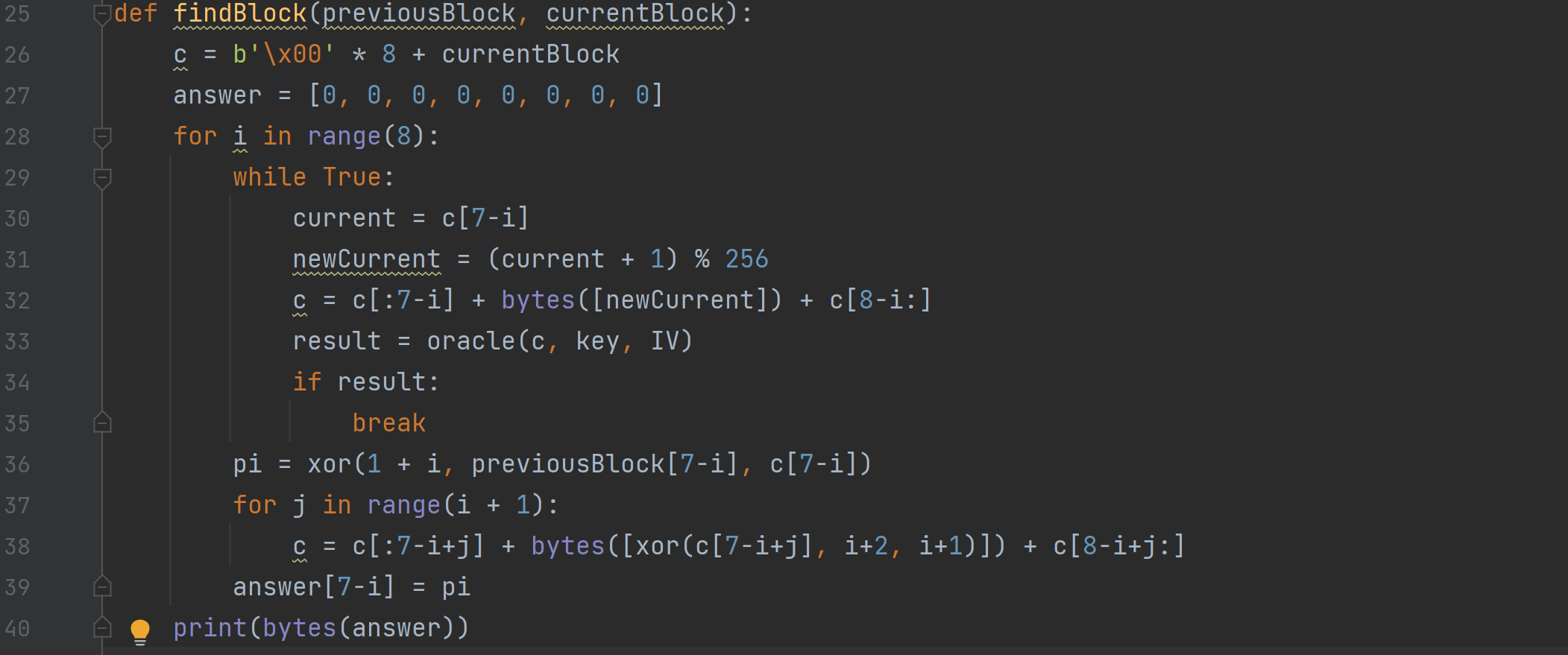
לבסוף נדפיס ואכן:  


**שלב 12:**ראשית נראה כי אכן אם "נמיר" את זה לבלוק הראשון זה יעבוד גם עליו:

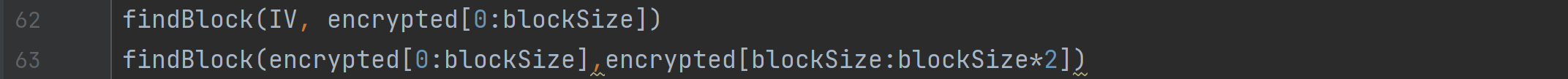


רק נחליף את מה שמצד ימין לשוויון במה שנחליף ונשמור על השמות אותו הדבר:  
בשורה 42: החצי הראשון של הencrypted יהיה הIV שלנו,  
בשורה 43: ניקח את החצי הראשון של הencrypted, והשאר נפעיל אותו הדבר על הc החדש שלנו ואכן:

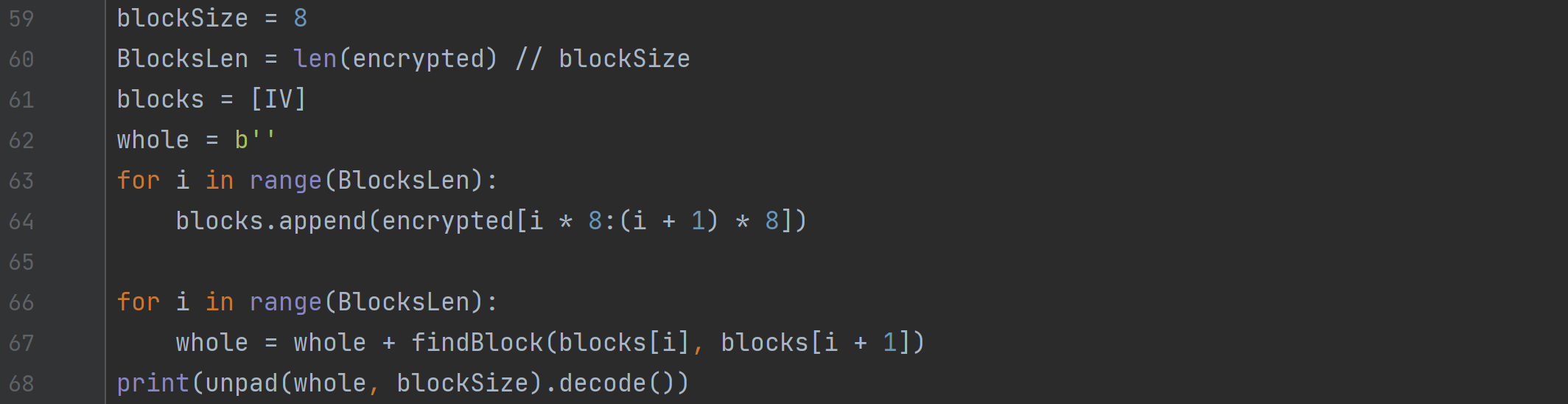
!! איזה כיף 😊

כעת נכניס את זה לפונקציה כללית:  


ונבדוק:



ואכן:  


כעת נעבור לכמעט סופי:  


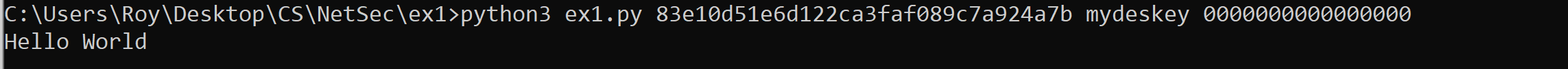
ואכן:

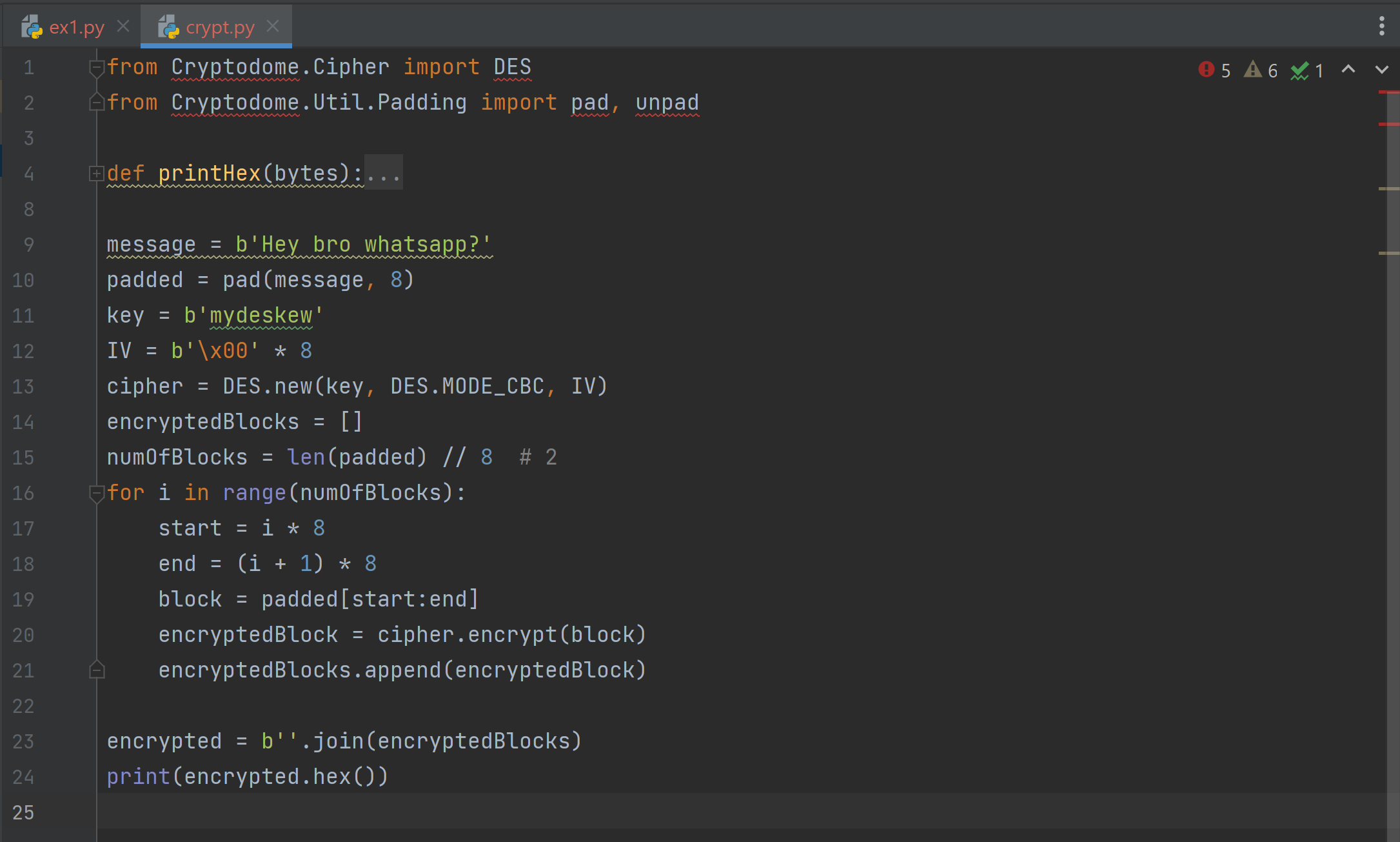


וכעת כל שנותר זה לקבל את הארגומנטים מהטרמינל:



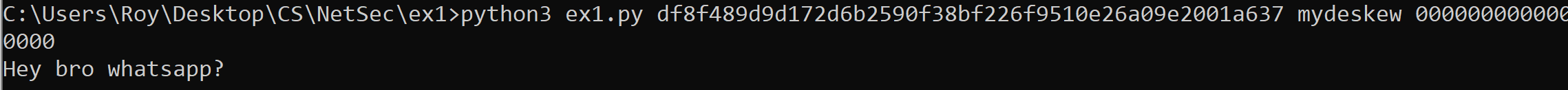
ובגלל שרשמתי את אותם השמות, מכאן זה כבר אותו הדבר, והידד:



נבדוק בשביל הכיף בדיקה נוספת:  


נריץ בשביל לראות את ההצפנה של זה:



נעתיק ונדביק לקוד שלנו ואכן:  


😊😊😊😊😊😊😊😊😊😊