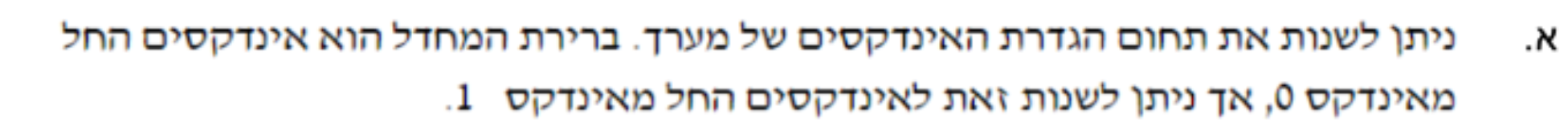
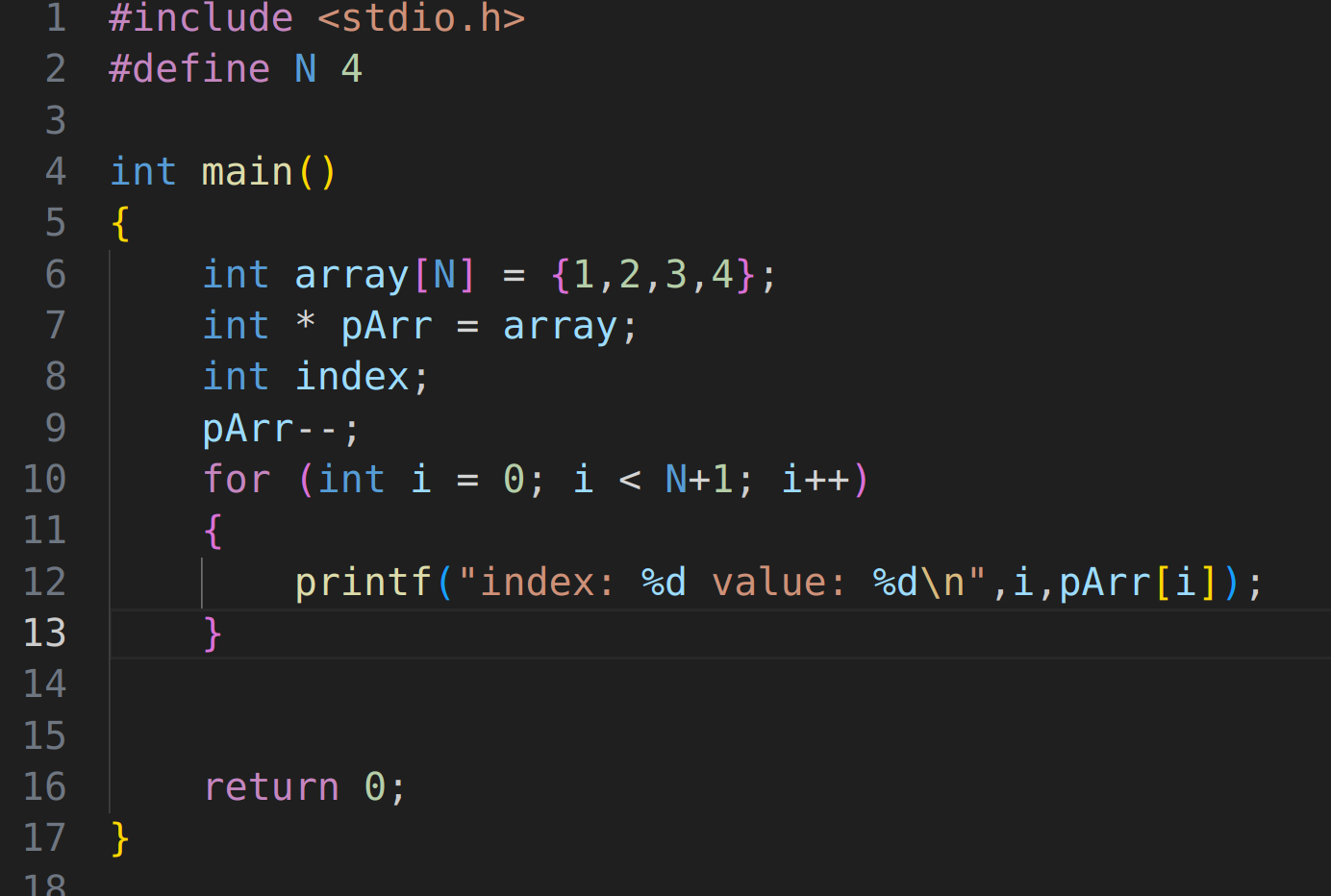
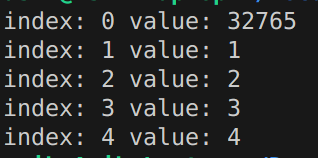
1.**נכון**- "אינדקס" במערך זה אופסט (offset) מסוים מנקודה בזיכרון, תכנית אפשר להגדיר מערך ולקחת כל נקודה כ"ההתחלה" של המערך ולפי כך יחסית למיקום הנבחר אפשר לשנות את המיקום היחסי של איברים של המערך. אבל שיטה זהו לא משנה את המיקום האמיתי של המערך או את האיברים שלה, ויותר מזה חלק מהמערך עלול ליהות לא משומש ויש סכנה לכתיבה לתוך זיכרון חיצוני.



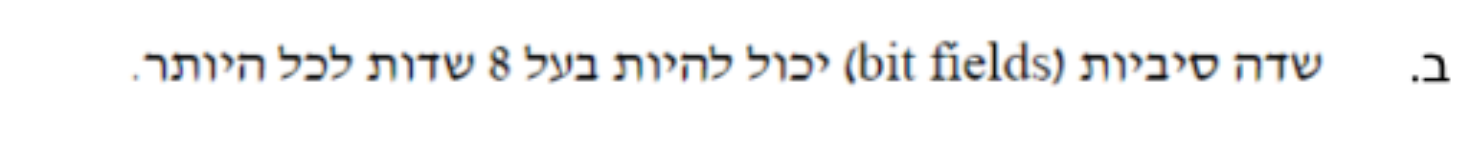
בקוד הזה אנחנו מגדירים מערך בעל 4 תווים. התו הראשון נימצא באינדקס "0".

אפשרי לקחת פוינטר למערך לקחת ולקחת אותו צד אחורה בזיכרון כך האינדקס הראשון יהיה באופסט של 1.

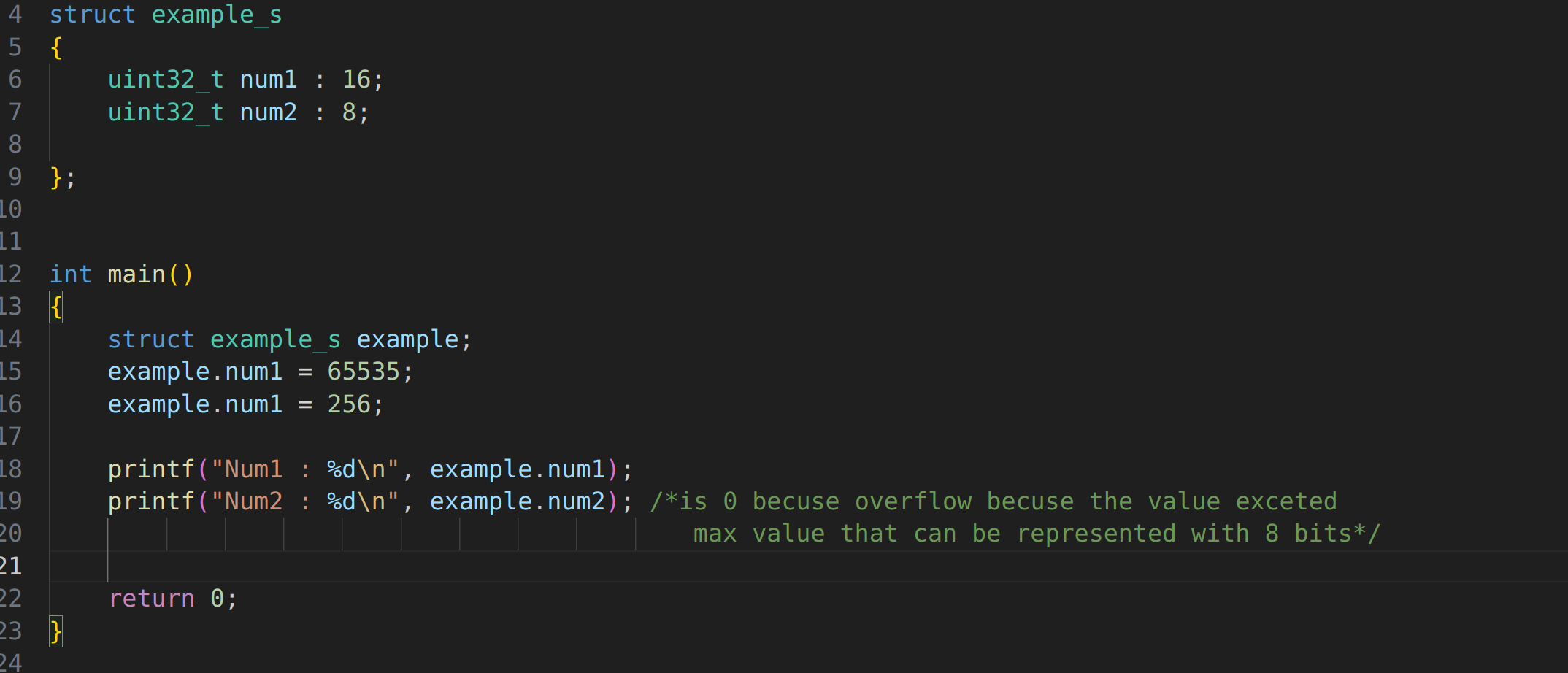
עם נעשה כך הערכים של המערך יהיה בין בינדקס 1-5

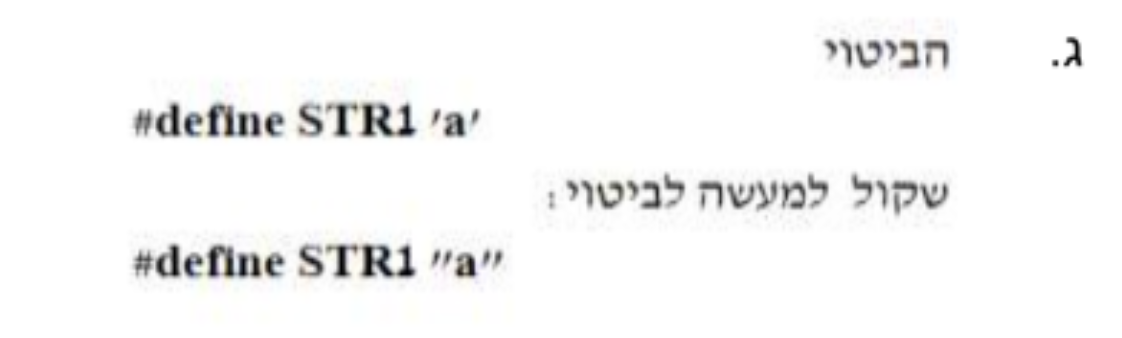
ואפשר לטעון ששיננו את הגדרת האינדקסים. למרות זהות שווה ניתן להגיד כי המערך עצמו לא הישתנה ואנו סתם עוברים על המקומות בזיכרון. התשובה תלויה בהגדרה של מה זה המערך – הפוינטר שמצבי על האיבר הראשון או איברים כולם כblock של משתנים בזיכרון.

2.

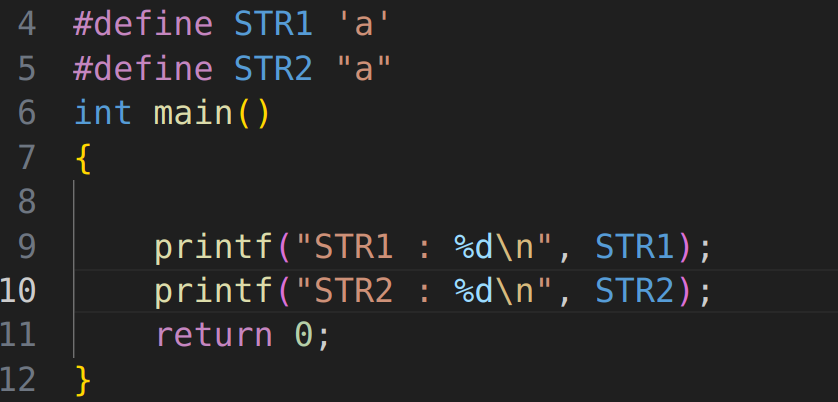


**לא נכון** , שדה סיביות יכול ליהות כל גודל המוגדר מראש כל עוד הוא קטן מכמות הביטים במשתנה.

****

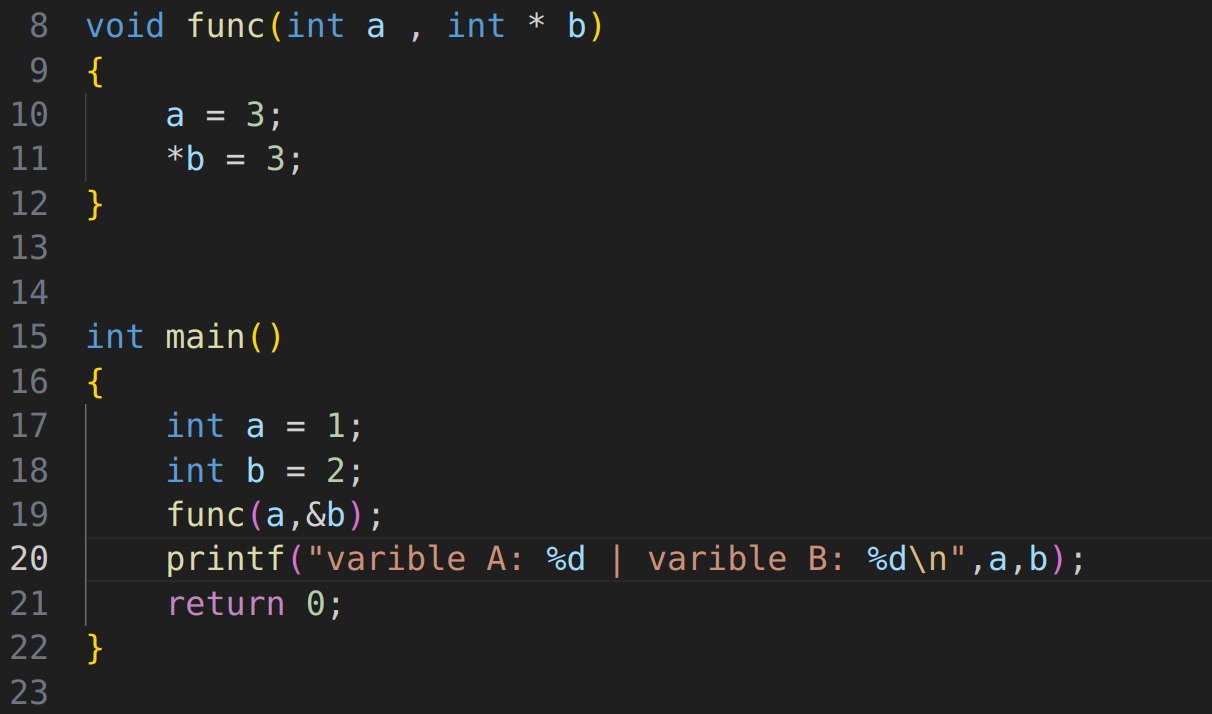
****

3. **לא נכון** , במקרה הראשון 'a’ הוא ערך של משתנה מסוג char בעל ערך assci המתאים לאותו תו. המקרה השני "a" הוא בנוי משתי תווים – תו 'a’ ותו '0\’.

ב STR1 שווה ל 97 שזה ערך של תו 'a’

ב STR2 הערך שווה למיקום בזיכרון של הפיונטר

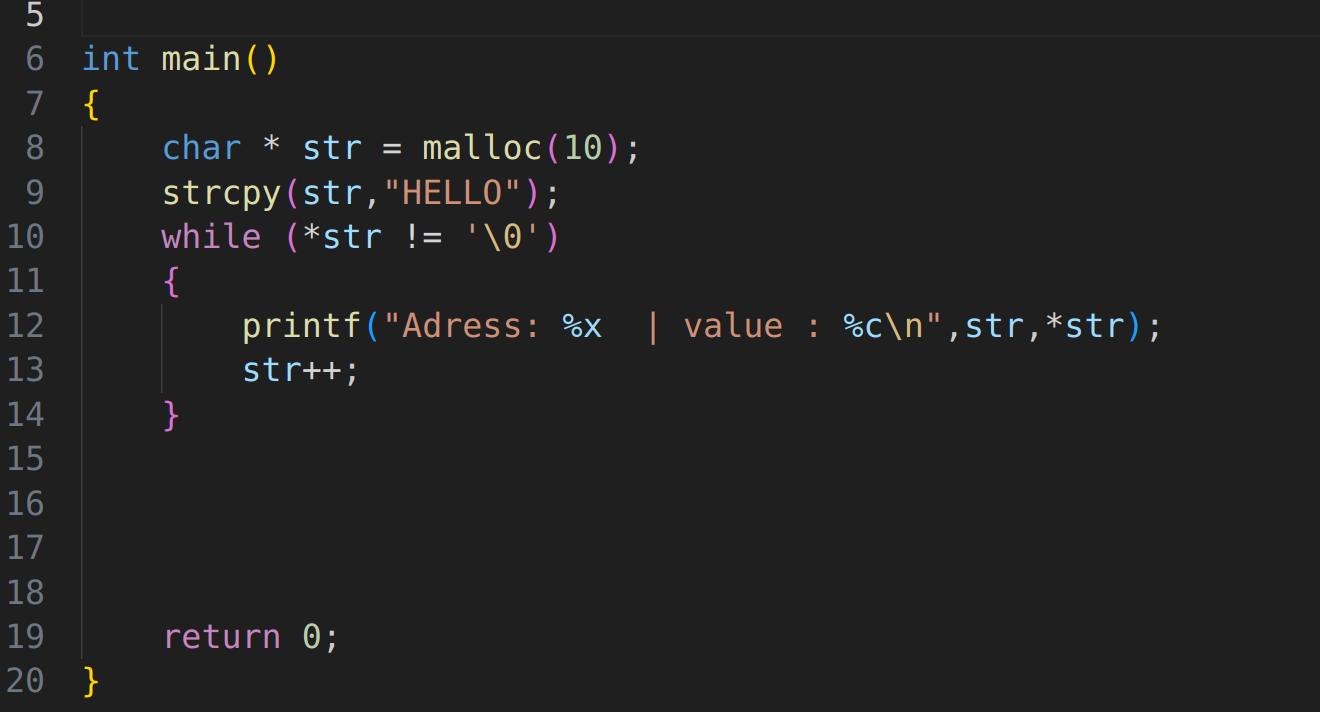
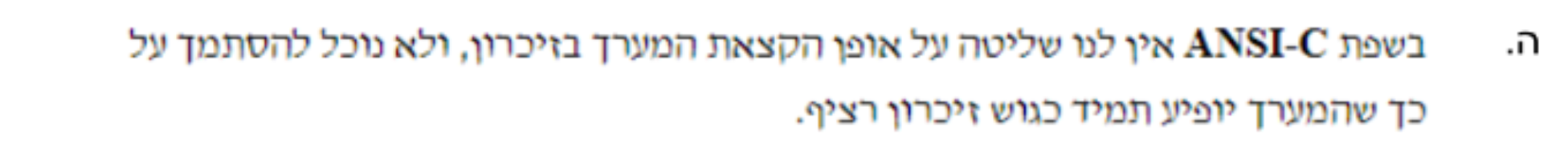
לאותו מערך.

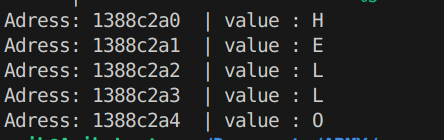
4. **לפעמים נכון ולפעמים לא נכון**, הפונקציה לא יכולה ישירות לשנות את הערכים שהעוברו כפרמטרים אבל היא כן יכולה לקבל פוינטר למקום בזיכרון לשנות את המקום הזיכרון הזה, כלומר ניתן לשנות את המשתנים שהעוברו by referance ולא by value.

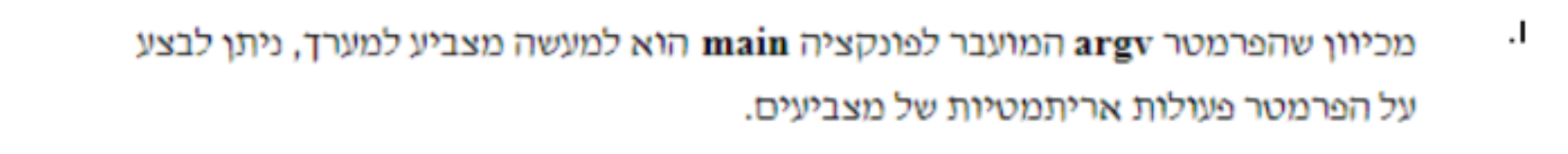
****

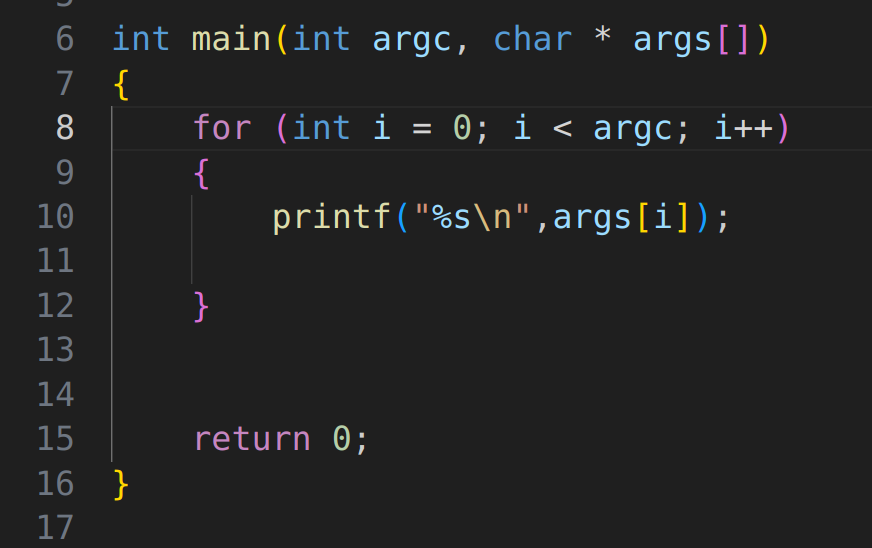
בדוגמה יש ערך a ו b

את שני הערכים מעבירים לפונקציה ובודקים עם עם הם ישתנו. ההבדל בין a ל b הוא שמעברים את a כערך ואת b כפוינטר. הערך שb מצביא עליו משתנה, לעומת זהות a החדש שבתוך הפונקציה נמחק בסיום ריצת הפונקציה.

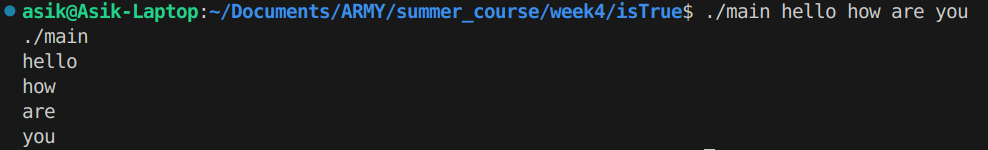
5. **לא נכון** , בשפת C המערכת הפעלה שולטת על ההקצה הזיכרון ואין לנו שליטה על איזה זיכרון מוקצה. למרות זהות כאשר אנחנו מקצים מערך אנחנו תמיד מקצים רצף של זיכרון רציף בגודל הנדרש. במקרה ונקצה זיכרון לכל איבר בזיכרון בנפרד המערכת הפעלה לא מיתחייבת שאיברים יהיו שמורים בזיכרון.

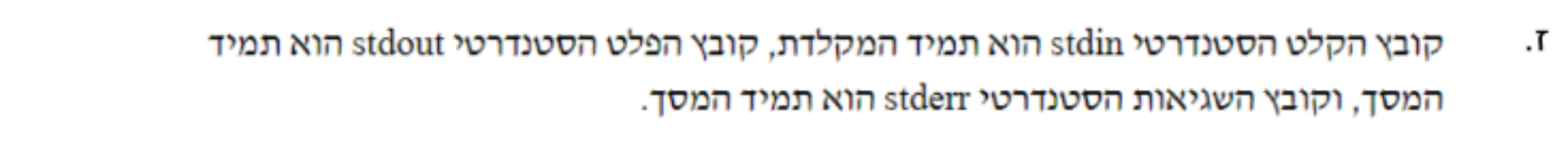
כפי שראוים בדוגמה הקצנו זיכרון וכל תו הוא התו במילה הוא תא הבא בזיכרון.

6. **נכון**,argv הוא מערך של מילים וניתן לעבור על כל מילה בעזרת argc שזה כמות המילים.



בדוגמה עוברים על כל מילה ומדפיסים אותה.

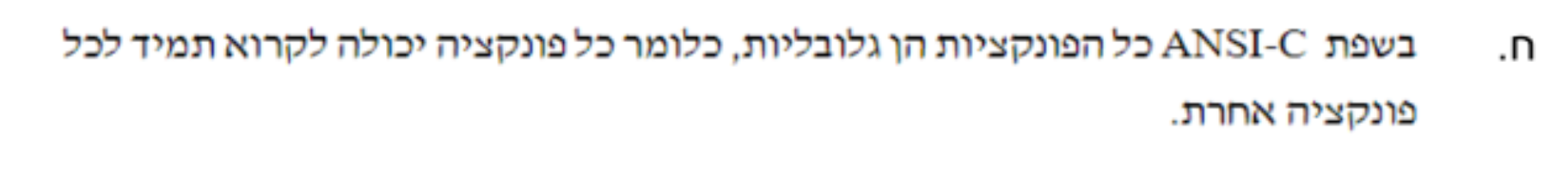
****

7. **לא נכון** , אפשרי להגדיר כל אחד מהם לכתוב גם לתוך קובץ המוגדר בהרצה התוכנה.

פקודה זהו כותבת את הפלט מהתוכנית הקודמת לתוך קובץ file.txt

עם הקובץ לא קיים הפקודה יוצרת אותו.

8.

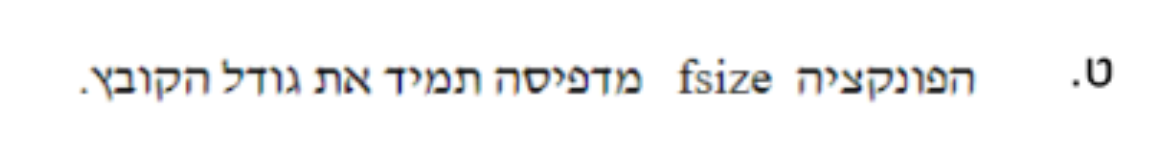


**לא נכון**, ב C יש חשיבות איפה כל פונקציה מוגדרת, למשל חייבים להגדיר על פונקציה לפני שמשתמשים בה.

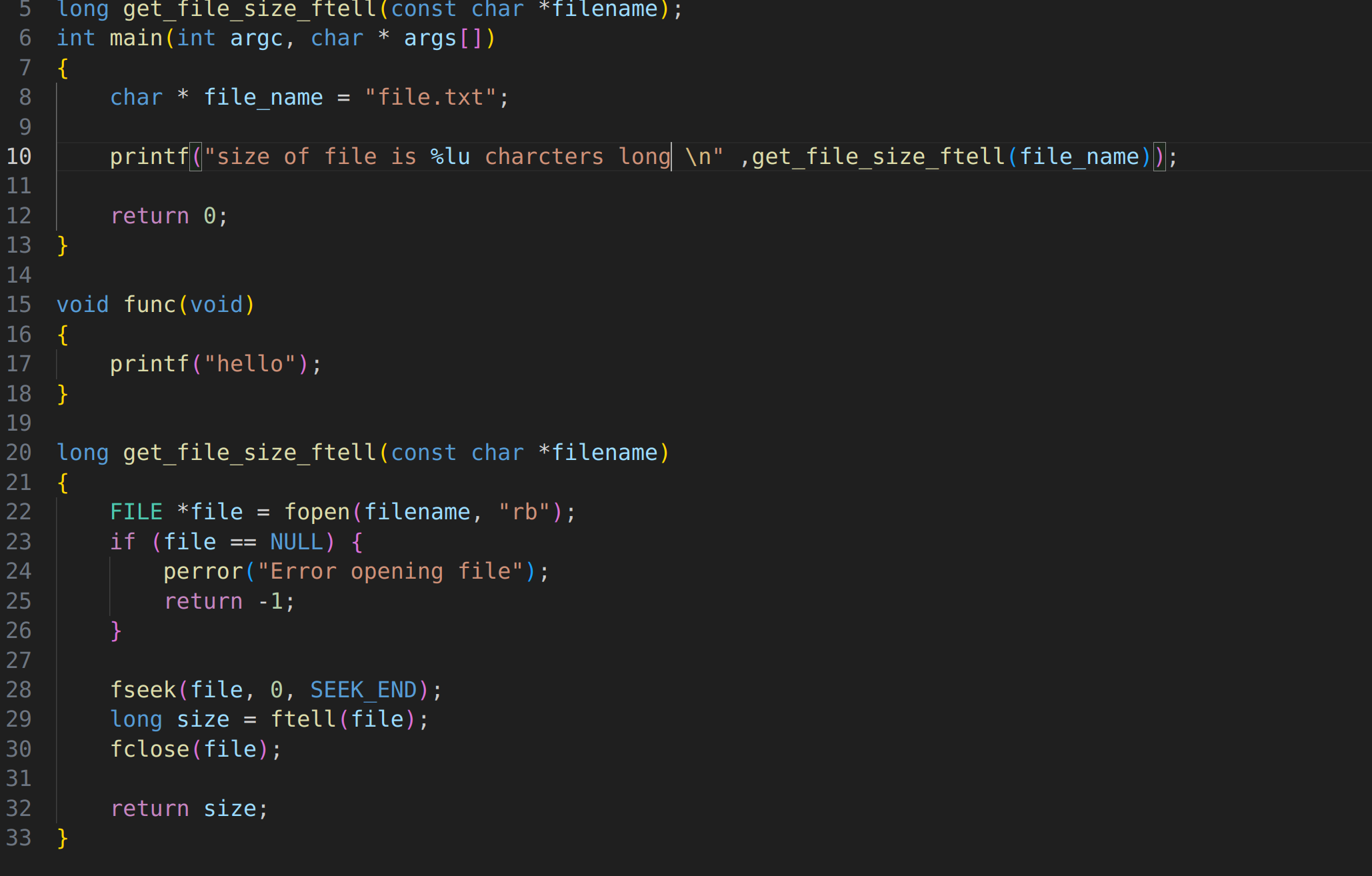
****

הדוגמה זהו הפונקציה מוגדרת אחרי הקריאה לה והתוכנית מחזירה טעות בקומפילצייה.

****

****

9. **לא נכון** , לא קיימת פונקציה כזאו. כדי לקבל את גודל הקובץ ניתן להישמש בfseek ו ftell בצורה כזהו :

עם מרצים על קובץ file.txt שיצרנו

לפני התוכנית תדפיס את אורך הקובץ

(כמות תווים).