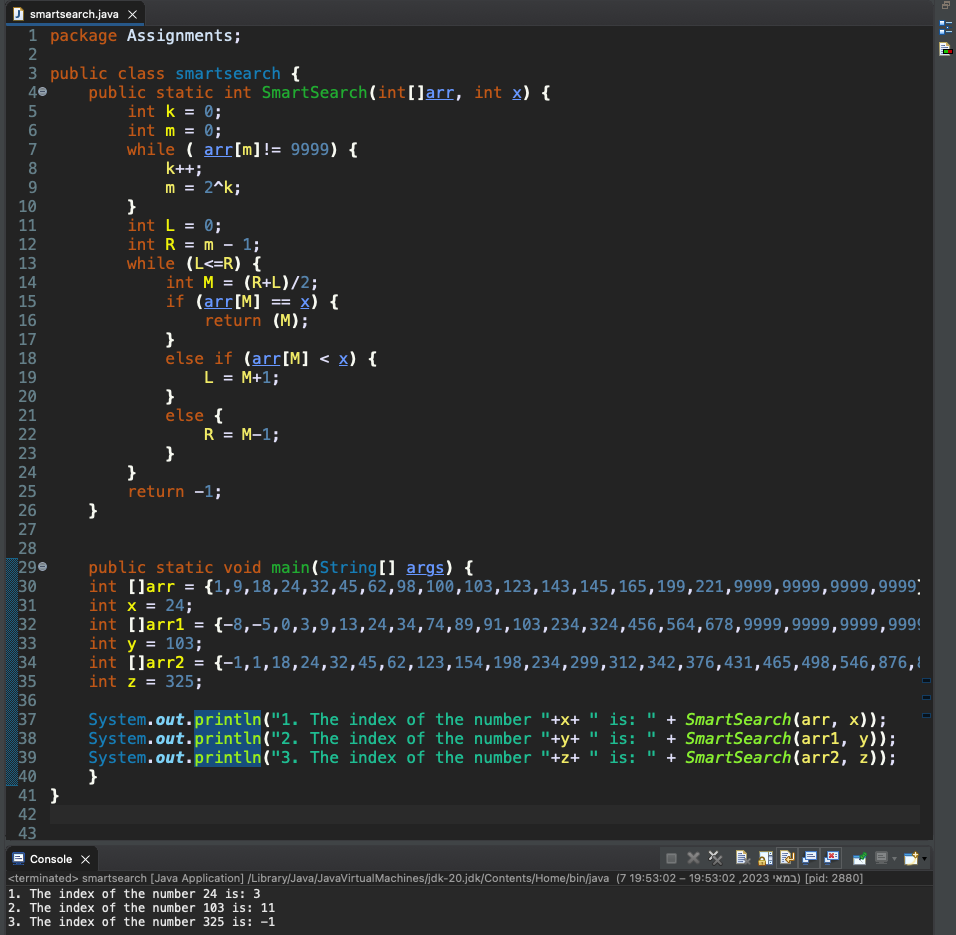
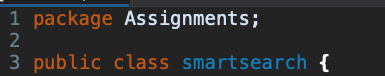
**מטלה 1 – חיפוש מספר במערך ממוין:**

**שם: אלעד טייר   
ת״ז: 318784097**

**תוצאות הריצה:**

****

**פירוט אודות התוכנית:**

**שורות 1-3:  
**

נפתחה תיקיה בשם Assignments .  
בתוכה נפתחה מחלקה בשם smartsearch.

**שורה 4:**

****

בתוך המחלקה יצרתי את הפונקציה – SmartSearch המקבלת נתונים של מספרים שלמים. הפונקציה תקבל מהמשתמש רשימה ״arr״ של מספרים שלמים.  
ומספר שלם – x שאותו תחפש בתוך הרשימה.

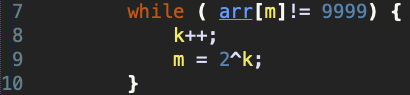
**שורות 5-6:  
**

יצירת משתני עזר מסוג של מספרים שלמים, ופעולת השמה לתוכם:  
**k** – יהיה משתנה סכימה ולכן יקבל את הערך 0.

**m** – ישמש כמשתנה לאורך הרשימה עד לאיבר הפיקטיבי (9999) לכן יתחיל מ-0.

סיבוכיות זמן ריצה: (2)O

**שורות 7-10:**



יצירת לולאת while שתרוץ כל עוד הערך של האיבר במיקום ה-m ברשימה   
יהיה שונה מ-9999.

במידה ומתקיים, בתוך הלולאה יתבצעו 2 פעולות:

* העלאת משתנה הסכימה k ב-1 בכל ריצה.
* והמשתנה m יוגדר מחדש בכל ריצה כך שבכל ריצה,   
  יגדיל את ערכו ב2 בחזקת ה-k העדכני לריצה , ().

משמעות הלולאה זו היא למצוא את המיקום בו מתחילים להתקיים ערכים פיקטיביים,   
בסיום הריצה (כאשר יגיע למספר שהוא 9999) המשתנה m יחזיר את המיקום האחרון הרלוונטי, זהו משתנה אקספוננציאלי. (...1,2,4,8,16,32,64)  
סיבוכיות זמן ריצה: ( log(m) )O

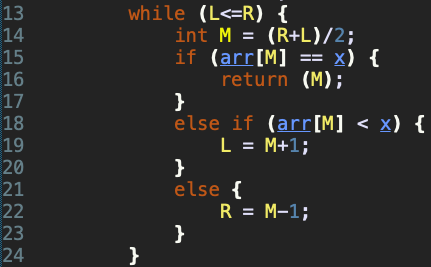
**שורות 11-12:**

****

יצירת 2 משתני עזר:

**L** – ישמש לערך השמאלי של הרשימה ולכן יקבל את המספר 0,   
מיקום האיבר השמאלי ביותר תמיד יהיה 0.  
**R** – ישמש בתור הסמן הימני ולכן יקבל את הערך של m פחות אחד.   
זאת אומרת, שמשתנה זה יקבל את מספר האיברים בהם יש רוב משמעותי   
של מספרים ממשיים.  
סיבוכיות זמן ריצה: (2)O

**שורה 13-24:**



יצירת לולאה מסוג while אשר תרוץ   
כל עוד הסמן השמאלי (L) קטן או שווה מהסמן הימני (R).   
נבצע הגבלה זו בכדי לא לרוץ על איברים שכבר רצנו עליהם במערך.

בתוך הלולאה:

* משתנה עזר M שיהווה כסמן לאיבר האמצעי במערך.  
  כלומר, הסמן M יהיה תלוי בסמן הימני והסמן השמאלי, כך שבמידה ויחול שינוי באחד מהם ערכו ישתנה.
* תנאי (if) – אשר בודק אם הערך במיקום M שווה למספר שקיבלנו מהמשתמש לבדוק (x),
  + **אם כן מתקיים,** יחזיר את הערך M ז״א המיקום וייצא מהפונקציה.
* אם לא (else if) – במידה והערך של האיבר במערך במיקום M יהיה קטן מהמספר שקיבלנו לבדוק אז יזיז את הסמן השמאלי L למיקום אחד אחריי M.
* אם לא (else) – במידה והערך של האיבר במערך במיקום M יהיה גדול מהמספר שקיבלנו לבדוק אז יזיז את הסמן הימני R למיקום אחד לפני .M

לולאה מסוג זה מבצעת חיפוש בינארי.   
אלגוריתם החיפוש הבינארי יכול להתקיים במידה ומערך הנתונים ממוין,   
ולכן ניתן להסיק כי האיבר מימין יהיה גדול מהאיבר שמשמאל, ולהיפך.  
לולאה זו ״חותכת״ חלקים במערך בכל ריצה על מנת למצוא את המספר x.

סיבוכיות זמן ריצה: ( log(m) )O

**שורה 25:**

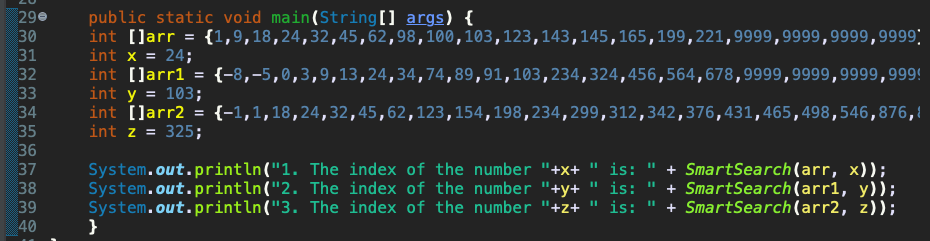


שורה זו מתקיימת רק אם במהלך הריצה של הלולאה לא נמצא המספר x שהתקבל מהמשתמש לבדיקה.  
המספר 1- ישמש כהגדרה לערך x שאינו קיים במערך.

פעולה זו קיימת בכדי שהפונקציה לא תהיה אינסופית ובמידה ועברה על כל המערך   
והערך x לא יהיה בו עדיין הפונקציה תחזיר תשובה.

סיבוכיות זמן ריצה: (1)O

**שורה 29-40:**



נפתח שגרת בדיקה לצורך הדפסה ושימוש בפונקציה SmartSearch.

* 3 משתני מערכים בשמות arr, arr1,arr2, מסוג של מספרים שלמים.
* 3 משתני עזר בשמות x,y,z של מספרים שלמים שישמשו אותנו בחיפוש המספר במערכים.

|  |  |  |  |
| --- | --- | --- | --- |
| המערך | arr – 20 איברים | arr1 – 30 איברים | arr2 – 40 איברים |
| מספר שלם לחיפוש במערך | x = 24 | y = 103 | z = 325 |

בשני המערכים הראשונים האיברים נמצאים במערך,  
במערך השלישי (arr2) האיבר z לא קיים במערך ולכן יחזיר 1-.

ההדפסה מורכבת ממחרוזת, מהמספר שיש לחפש (x) ומפונקציית SmartSearch:



**ניתוח סיבוכיות זמן הריצה:**

בעקבות כך שהערך של המקדמים (2) והקבועים (5) זניחים ניתן לקבוע כי,  
**סיבוכיות זמן הריצה של הפונקציה SmartSearch =**