**שיעורי בית 11: יעילות**

**הנחיות**

* בתרגיל זה יש מספר פעמים בהן אתם מתבקשים לחשב את סיבוכיות זמן הריצה של פונקציה. יש לפרט זאת בהערה בקוד.
* **לשאלות 3, 4, 5 נתון לכם שלד הקובץ intLinkedList.c** – כדאי להשתמש בו.
* **שימו לב:** המטרה שלנו היא לא לכתוב רק קוד נכון (זה כבר קטן עלינו) אלא קוד יעיל (זה כיף אמיתי!). לכן חישבו על הפיתרון האיכותי והיפה ביותר!
* **טיפ של אלופים:** אפשר להתחיל מהפתרון הראשון שקופץ לראש (בדרך כלל יעבוד טוב אבל לא יהיה הכי יעיל) ולחשוב איך אפשר לשפר אותו מבחינת סיבוכיות זמן הריצה.

**שאלות**

1. כתבו פונקציה יעילה ככל הניתן, המקבלת מספר חיובי שלם n ומדפיסה את כל המספרים הקטנים או שווים לו שהשורש שלהם הוא מספר שלם. חשבו את זמן הריצה של הפונקציה (שימו לב, ישנו פיתרון ב-(O(n ויש פיתרון מהיר יותר!).

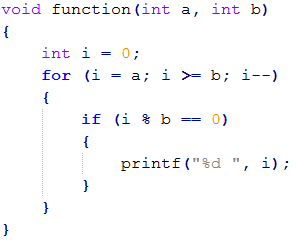
כתבו main העושה בפונקציה שימוש.

דוגמת ריצה:



הנחיה: יש להוסיף getchar גם לאחר קליטת המספר ב-main וגם בסופו.

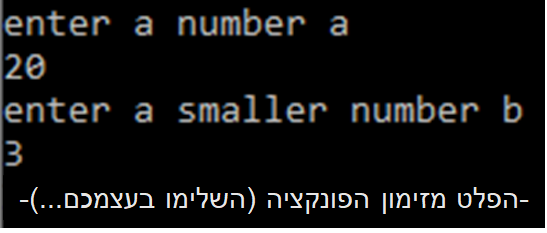
2. נתונה הפונקציה הבאה, המקבלת שני מספרים שלמים חיוביים a, b (ניתן להניח ש-a>b):



* מה מבצעת הפונקציה?
* מהי סיבוכיות זמן-הריצה של הפונקציה?
* כתבו פונקציה יעילה יותר (מבחינת מספר איטרציות של הלולאה).

כתבו main העושה שימוש בפונקציה

דוגמת ריצה:



הנחיה: יש להוסיף getchar לאחר קליטת כל אחד מהמספרים ב-main וגם בסופו.

ה-main צריך לזמן את הפונקציה המעודכנת שכתבתם.

**יש לדאוג שההדפסה תהיה בסדר עולה.**

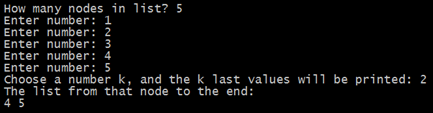
3. כתבו פונקציה בשם concatenate המקבלת 2 מצביעים לרשימות מקושרות (שמכילות int כ-data), ומחזירה מצביע לראש רשימה המכילה את כל האיברים של 2 הרשימות משורשרות. על סיבוכיות זמן הריצה של הפונקציה לא לעלות על אורך הרשימה הקצרה יותר ועוד מספר קבוע של פעולות.

4. כתבו פונקציה המקבלת מצביע לחוליה ראשונה של רשימה מקושרת של מספרים, ומספר שלם k ומחזירה מצביע לאיבר ה-k מהסוף ברשימה. לאחר מכן הדפיסו את הרשימה מהמקום ה-k מהסוף, ועד הסוף.

ניתן להניח שהרשימה מכילה לפחות K איברים.

* נסו לעבור על הרשימה רק פעם אחת (רמז: חלקו ל-2 לולאות).
* חשבו את זמן הריצה של הפונקציה כתלות בגודל הרשימה (נקרא לו n) ו-k.

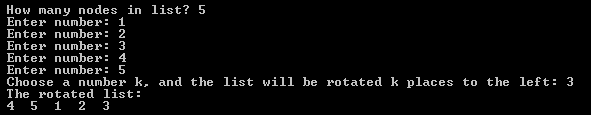
דוגמת ריצה:



5. כתבו פונקציה המקבלת מצביע לרשימה מקושרת של מספרים ומספר שלם k, ומסובבת את הרשימה k מקומות שמאלה.

* אין לשמור עותק נוסף של הרשימה.
* בכמה מקומות בתוכנית אתם משנים את ה-next של חולייה כלשהי? (ניתן לפתור עם מספר שינויים שאינו תלוי באורך הרשימה).

דוגמת ריצה:



6. **בונוס:** היכנסו לאתר של פרויקט אוילר והירשמו. מומלץ גם לקרוא עליו קצת :)

https://projecteuler.net/about

כל התרגילים בפרויקט אוילר ניתנים להרצה על מחשב סביר בפחות מדקה (אם מתכננים אלגוריתם יעיל...).

פתרו את 2 התרגילים הראשונים ב-archive בדרך הכי יעילה שאתם יכולים.

צרפו את הקוד/הסבר על הפתרון וחישוב של זמני הריצה.

דוגמת ריצה לתרגיל הראשון:



דוגמת ריצה לתרגיל השני:



7. **בונוס:** מהו זמן הריצה של חיפוש בינארי? הסבירו את ניתוח זמן הריצה. במידה ונעזרתם באינטרנט, ציינו זאת ותנו קרדיט.

**בהצלחה!**