**מנמ"א - ממן 14**  
204122188

**שאלה 1**ראשית נסמן:  
 - "מבנה הנתונים" אליו שייכים איברי הקבוצה הנתונה. .  
 - קבוצת פונקציות הגיבוב המדוברות..  
 פלט הפונקציות הוא מספר שלם בין ל .  
 – טבלת הביטים הנתונה. .

1. הסיכוי הוא 0. נסביר:  
   היות ועבור כל איבר בקבוצה מסומנים ב1 כל המקומות המותאמים לו בT ע"י פונקציות ,  
   לא יתכן כי בעת חיפוש ימצא באחד המקומות האלו 0 (זאת בהנחה שאין פונקציית מחיקה),  
   ולכן הבדיקה שלו תחזיר תמיד אמת.
2. לפי ההדרכה ראשית נחשב  
   מה הסיכוי שעבור איבר כלשהו (שלא בהכרח ב ),  
   קיימת פונקציה מסויימת, , שתחזיר עבורו מספר, (שהוא אינדקס מסויים ב),  
   כך שלכל וכל מתקיים .  
     
   כלומר- היות ול יש סיכוי של להתקבל ע"י כל אחד מפעולות   
   נחשב מה הסיכוי שכל אחת מפעולות על כל אחד מאיברי נתנה תא שאיננו :  
     
   הסיכוי שתא זה לא נבחר עבור ריצה מסויימת של הוא .  
   בדיקה זו צריכה להחזיר **לכל** הריצות של כל הפונקציות על מסיים איבר שונה- ולכן:  
   לכל איבר יש סיכוי של לא להיות מנותב ע"י אף פונקציה אל .  
     
   היות ותנאי זה נדרש עבור כל , נדרש שנחזור על התוצאה הנ"ל פעמים,  
     
   כך שבסה"כ הסיכוי שתא : .   
     
   כעת, הסיכוי שאיבר שאיננו במבנה הנתונים ()  
   יקיים לכל .  
   אז לפי טענת העזר הסיכוי שעבור מסויים- לא יתן תא שיש בו הוא היות והחיפוש דורש שעל מנת שאיבר יוכרז כאיבר ב בכל החיפושים יתקבל עלינו לחזור על תנאי זה עבור כל הפונקציות  
     
   לכן בסה"כ הסיכוי הוא . משל.
3. לפי סעיף ב' התשובה היא:

תיעוד כללי:

ראשית נסביר את מהות התכנית.

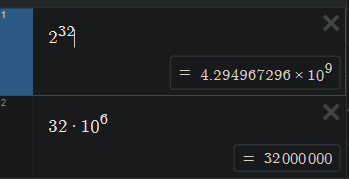
מטרתה של תכנית זו היא לממש מבנה נתונים, בו יוכנסו איברים הנקלטים ע"י. האיברים יוכנסו למבנה הנתונים פונקציות גיבוב.

במקרה שלנו, ישנה טבלה T בגודל m, קובץ בן N איברים ו-K פונקציות גיבוב מספריית murmurhash3.

פונקציה הראשית:

1. קוראים לפונקציה take\_inputs אשר קולטת מהמשתמש את m ואת K.
2. קוראים לפונקציה zero\_list\_maker המאתחלת את הטבלה ל '0'ים.
3. בעזרת קריאה לפונקציה pars\_file מחלצים את N האיברים מקובץ הקלט, אותם עתידים להכניס לטבלה.
4. בעזרת קריאה לפונקציה inserts מכניסים את אותם N איברים אל הטבלה ע"י K פונקציות הגיבוב murmurhash.
5. בעזרת קריאה לפונקציה pars\_file מחלצים את הקלטים אותם אנו רוצים לבחון מקובץ הקלט של הבדיקה.
6. קוראים לפונקציה check ובודקים עבור כל איבר מאיברי הבדיקה האם הוא שייך או איננו שייך למבנה הנתונים.

\*פירוט וזמני ריצה מפורטים בתכנית עצמה בראש כל פונקציה.

\*\*פירוט על הפונקציה inserts:  
כמות המספרים שhash יכול לייצר (נסמנה A)  
לכמות המספרים שT שלנו יוכל לקלוט. (נסמנה B)  
  
וברור כי . לכן- על מנת שלא נקבל ערכי hash הנמצאים  
מחוץ לטבלה אך בכל זאת נשמור על פיזור אחיד ככל האפשר-  
נעזר בפונקציית מודולו על כל פלט של פונקציית hash.

**הוראות הרצה**

כדי להריץ את התכנית הראשית יש להכין קבצי 2 קבצי .txt :

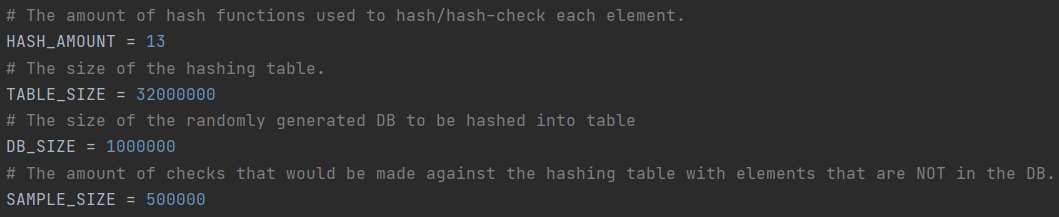
1. קובץ האיברים אותם רוצים להכניס לטבלה בשם DB\_FILE.
2. קובץ איברים אותם רוצים לבדוק האם שייכים לטבלה בשם CHECK\_FILE.
3. להריץ ולראות את הפלט המתאים.

כדי להריץ את תכנית הסטטיסטיקות:

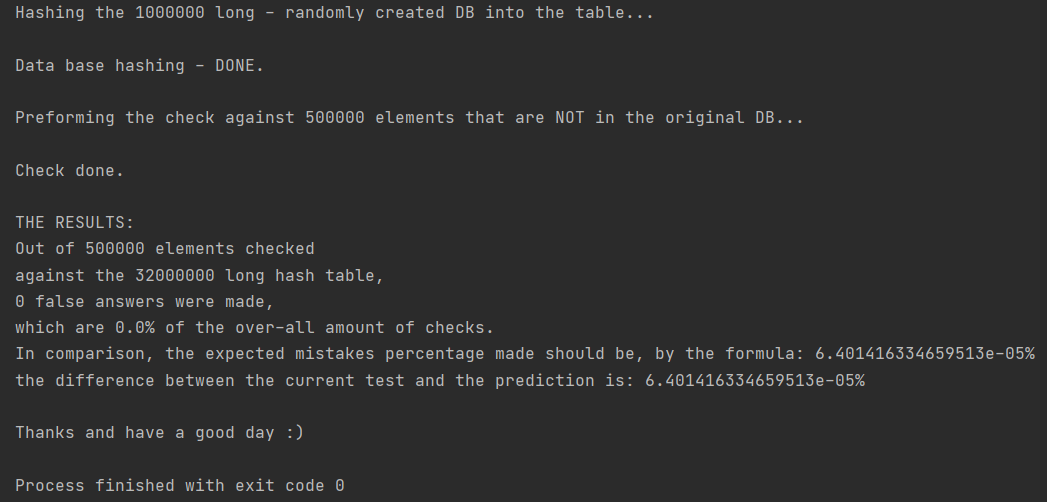
1. ראשית יש להביט במשתנים הקבועים שהם:
   1. כמות פונקציות הגיבוב - HASH\_AMOUNT
   2. גודל הטבלה - TABLE\_SIZE
   3. כמות האיברים אותם רוצים להכניס לטבלה - DB\_SIZE
   4. כמות האיברים אותם רוצים לבדוק האם שייכים לטבלה – SAMPLE\_SIZE
2. כפריסט, הנתונים מאותחלים בדומה לנתוני השאלה.
3. להריץ ולראות את הפלט המתאים.

**דוגמא להרצה**

נתוני ההרצה של פריסט הנתונים:



הפלט:



ישנה אפשרות לשמתמש לשנות את הקבועים כדי לראות תוצאות שונות כרצונו.