# בס"ד

# תרגיל 2: (טבלאות גיבוב)

בתרגיל זה, מטרתנו לממש מערכת עדכנית לשמירת מידע בעזרת בטבלאות גיבוב. (מערכת הפונים והמתנדבים) לשם כך נחזיק 2 טבלאות: אחת מכילה את **רשימת המתנדבים** במערכת, כאשר המפתח הוא שם המתנדב, והשנייה את **רשימת הפונים** למוקד, כאשר המפתח יהיה מספר הטלפון ממנו התקשרו אל המוקד.

בעזרת מבנה נתונים זה נרצה לכתוב מערכת שמאפשרת את הפעולות הבאות:

* הוספה/מחיקה מתנדב מהמאגר.
* הוספת מקבל שרות(פונה) למאגר.
* בהינתן פונה, מי המתנדבים הגיעו אליו.
* בהינתן מתנדב, אצל אילו פונים הוא ביקר.

נגדיר את העבודה לפי השלבים הבאים :

**שלב א:**

נגדיר **טבלת גיבוב כללית בשיטת המיעון הפתוח**. לשם כך, כתבו מחלקה **מטומפלטת** **ומופשטת לייצוג** טבלת הגיבוב(מחלקת טמפלייט כללית לטבלת גיבוב). כאשר הערכים בטבלה יהיו מטיפוס כלשהו (מסוג T). ולכל איבר יהיה מפתח מסוג K (השתמשו ב- template כדי לאפשר T,K כללים).

ניתן להגדיר מחלקת עזר נוספת בשם **item** שמייצגת איבר בודד בטבלה ומכילה את הנתון T, את המפתח K ושדה נוסף שמגדיר את המצב הנוכחי (מלא, ריק מחוק)

enum state {empty, full, deleted};

template <class T,class K>

class Item

{

public:

T data;

K key;

state flag;

Item(){}

Item(T d, K k, state f){ data=d; key=k; flag=f;}

};

**טבלת הגיבוב** צריכה להכיל לפחות את הפונקציות הבאות:

* **בנאי** (מקבל כפרמטר את גודל הטבלה ומאתחל טבלה בגודל מספר ראשוני הקרוב ביותר לגודל המבוקש).
* פונקציה **הורסת**
* פונקציה גיבוב h1 שמקבלת מפתח k מסוג K ומחזירה מיקום בטבלה
* פונקציית דילוג h2.

שימו לב - שהפונקציות h1,h2 תלויות במידע שנשמר בטבלה ולסוג המפתח: לכן מלכתחילה הן וירטואליות טהורות ורק במחלקה היורשת, המתייחסת לסוג מסוים הן ימומשו.

* פונקציה hash(K,I) המחזירה את המיקום בטבלה עבור מפתח k בניסיון I.
* פונקציה לחיפוש איבר (אם האיבר לא נמצא מחזיר -1).
* פונקציה להכנסת איבר לטבלה
* פונקציה למחיקת איבר מהטבלה.
* פונקצית פלט להצגת תוכן הטבלה כולה.

בשלב זה נעצור ונבדוק שהכל עובד – איך?

נגדיר טבלת גיבוב hashint שמשתמשת במחלקה הנ"ל עם ערכים מטיפוס int ומפתח גם מטיפוס int ונממש את הפונקציות הנדרשות. לאחר מכן נכתוב תוכנית ראשית קצרה שמאפשרת להכניס, למחוק ולחפש מספרים בטבלה.

**רק אם הכל עובד כראוי נמשיך לשלב ב!**

**שלב ב:**

1. **הגדירו מחלקה volunteer לייצוג מתנדב בטבלת גיבוב:**  
   במחלקה נשמור את פרטי המתנדב: שם, כתובת מגורים, מספר טלפון (אפשר להתייחס אליו כאל מספר מסוג int אם מתעלמים מהאפס שמייצג את הקידומת), שם ישוב.

בנוסף נגדיר: בנאי, הורס, אופרטור השמה, אופרטור קלט, אופרטור הדפסה, ואופרטורים נוספים לפי הצורך (בפרט את האופרטור ==, שישווה מתנדבים על פי השם שלהם).

1. הגדירו מחלקה לייצוג טבלת גיבוב (שיורשת את hash<volunteer,string> שהוגדרה בשלב א') שמכילה את כל המתנדבים, כאשר המפתח הוא שמו של המתנדב (מסוג string). הגדירו מחדש רק את הפונקציות הדרושות

**שלב ג:**

1. **הגדירו מחלקה לייצוג פונהclient** , כלומר משתמש שנעזר בארגון:

השדות במחלקה יהיו- שם, מספר טלפון, כתובת, ורשימת שמות המתנדבים שסייעו לו עד כה – ללא כפילויות. והמפתח של פונה יהיה מספר הטלפון של הפונה (אפשר להתייחס אליו כאל מספר מסוג int אם מתעלמים מהאפס שמייצג את הקידומת).

הוסיפו למחלקה זו פונקציות ואופרטורים לפי הצורך (>>,<<,== , = וכו').

ב. הגדירו מחלקה לייצוג טבלת גיבוב (שיורשת את הטבלה hash<client,int> שהוגדרה בשלב א') שמכילה את כל הפונים לארגון, כאשר המפתח הוא מספר הטלפון של הפונה (מסוג int). הגדירו מחדש רק את הפונקציות הדרושות.

**שלב ד:**

**הגדירו מחלקה לייצוג המאגר כולו. השדות במחלקה זו יהיו:**

* טבלת גיבוב עבור רשימת המתנדבים, על פי שם של המתנדב
* טבלת גיבוב עבור הפונים לארגון, על פי מספר הטלפון של הפונה

על המערכת לאפשר את הפעולות הבאות:

1. אתחול טבלאות גיבוב בגודל 100 כ"א.
2. הוספה/מחיקה של מתנדב.
3. הוספה/מחיקה של פונה חדש למערכת.
4. הוספת מתנדב שסייע לפונה: יש לוודא שאמנם מתנדב זה קיים במערכת.
5. בהינתן פונה יש להדפיס את רשימת שמות המתנדבים שסייעו לו. (כולל מתנדבים שכבר לא מתנדבים יותר, ובלי להדפיס שם של מתנדב יותר מפעם אחת).
6. בהינתן מתנדב, יש להדפיס את רשימת הפונים שהוא סייע להם. (זו פעולה יקרה שתדרוש לעבור על כל טבלת הגיבוב של הפונים. בונוס ינתן לעדכון התרגיל באופן שפעולה זו תתבצע ביעילות)

**הערה: גם במימוש היעיל, יש לקחת בחשבון שאם מחפשים זאת עבור מתנדב שכבר נמחק, הפעולה הזו לא! תתבצע בצורה היעילה.**

1. הדפסת טבלת המתנדבים וטבלת הלקוחות. יש לשים לב שיודפסו פרטי המתנדבים ופרטי הלקוחות. (ההדפסה לא תיבדק בבודק האוטומטי אלא בבדיקה ידנית בלבד).

**שלב ה:**

השתמשו בתוכנית הראשית המופיעה במודל, שמציגה למשתמש תפריט עם רשימת האפשרויות לעיל ומאפשרת לו לבחור בהן שוב ושוב עד לבחירת אפשרות סיום.

**בהצלחה!**

בס"ד

דוגמת הרצה:

Hash Tables  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
n  
Enter volunteer name phone address  
avraham 123 haifa  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
n  
Enter volunteer name phone address  
isaac 234 afula  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
n  
Enter volunteer name phone address  
yaacov 345 natania  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
c  
please enter name and phone and address of client  
reuven 1234 haifa  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
c  
please enter name and phone and address of client  
shimon 2345 jerusalem  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
c  
please enter name and phone and address of client  
levi 3456 afula  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
p  
Volunteers:  
isaac:volunteer name=isaac pnone=234 address=afula  
avraham:volunteer name=avraham pnone=123 address=haifa  
yaacov:volunteer name=yaacov pnone=345 address=natania  
Clients:  
1234:client name: reuven phone: 1234 address: haifa  
2345:client name: shimon phone: 2345 address: jerusalem  
3456:client name: levi phone: 3456 address: afula  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
l  
enter volunteer name and client phone avraham 1234  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
l  
enter volunteer name and client phone avraham 3456  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
l  
enter volunteer name and client phone isaac 1234  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
l  
enter volunteer name and client phone isaac 2345  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
p  
Volunteers:  
isaac:volunteer name=isaac pnone=234 address=afula  
avraham:volunteer name=avraham pnone=123 address=haifa  
yaacov:volunteer name=yaacov pnone=345 address=natania  
Clients:  
1234:client name: reuven phone: 1234 address: haifa helped by:  avraham isaac  
2345:client name: shimon phone: 2345 address: jerusalem helped by:  isaac  
3456:client name: levi phone: 3456 address: afula helped by:  avraham  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
\*  
enter client phone 123  
The volunteers that helped to client 123:  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
\*  
enter client phone 1234  
The volunteers that helped to client 1234:  avraham isaac  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
i  
enter volunteer name avraham  
The clients that were helped by volunteer avraham: reuven levi  
  
Choose one of the following  
n: New volunteer  
d: Del a volunteer  
c: New client  
l: Add a connection volunteer-client  
\*: Print volunteers that helped a client  
i: Print clients that were helped by a voluneer  
p: Print hash tables  
e: Exit  
e

Bye