**עבודה 2 – קבצים, מבנים ומערכים דינאמיים**

**מטרת העבודה:**

העבודה עוסקת בקריאה וכתיבה לקבצים ובשימוש במערכים דינאמיים.

מבני הנתונים שתבנו ישמשו אתכם כדי לענות על השאילתות המצוינות בתפריט.

במהלך העבודה תתבקשו לענות על כל אחת מהשאילתות שהציב המשתמש ולעדכן את הקובץ המתאים.

**תיאור העבודה:**

בעבודה זו עליכם לדרג את שיעורי הצפייה בסדרות טלוויזיה המסופקות ע"י 2 חברות NetChain ו- SeakNet.

כל אחת מהחברות האלו מספקות אוסף של סדרות טלוויזיה קלאסיות ומקוריות.

בעלי החברה מעוניינים לדרג את הסדרות שבבעלותם לפי מדד הצפיות של לקוחותיה.

נתוני הצפייה של הלקוחות נקלטים כזוגות של נתונים: קוד סדרה ומספר עונה.

לכל חברה נשמר קובץ המכיל את נתוני הסדרות, עבור כל סדרה נשמרת שורה ובה הנתונים הבאים:

* שם הסדרה – 20 תווים לכל היותר
* קוד הסדרה – 3 ספרות לכל היותר
* מספר העונות – 2 ספרות לכל היותר
* האם הסדרה פעילה – תו ('Y' או 'N')

**שימו לב,**

**בין נתון לנתון מופיע פסיק.**

**הקבצים ממוינים לפי קוד סדרה בסדר עולה.**

לדוגמה, הסדרה " Money Heist"

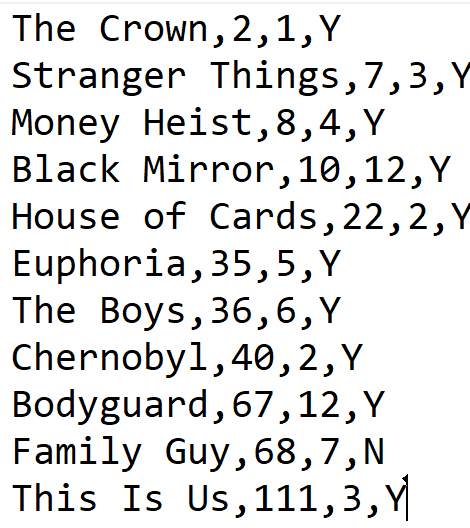
קוד הסדרה הוא 8

מספר העונות הוא: 4

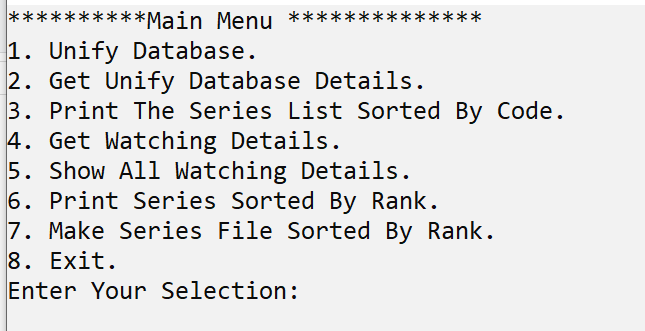
הסדרה פעילה

השורה המתאימה לסדרה זו בקובץ תיראה כך:

Money Heist,8,4,Y

להלן דוגמה לקובץ הסדרות:

עם הפעלת התוכנית יוצג התפריט הבא:



**אופציה 1 בתפריט: Unify Database**

אופציה זו חייבת להתבצע בהתחלה.

מטרתה היא לאחד את נתוני 2 הקבצים הממוינים לפי קוד סדרה, לקובץ אחד. המיזוג של 2 הקבצים יתבצע באופן הבא:

במידה והסדרה מופיעה רק באחד מהקבצים,

אם הסדרה לא פעילה => לא תיכתב שורה לקובץ

אם הסדרה פעילה => תיכתב שורה לקובץ עם נתוני הסדרה

במידה והסדרה מופיעה ב 2 הקבצים יש לטפל בכל אחד מהאפשרויות הבאות:

1. שתי הסדרות לא פעילות => לא תיכתב שורה לקובץ
2. שתי הסדרות פעילות => תיכתב שורה לקובץ כך שמספר העונות יהיה הגבוה מבין השניים.
3. רק סדרה אחת פעילה => תיכתב שורה לקובץ עם נתוני הסדרה הפעילה.

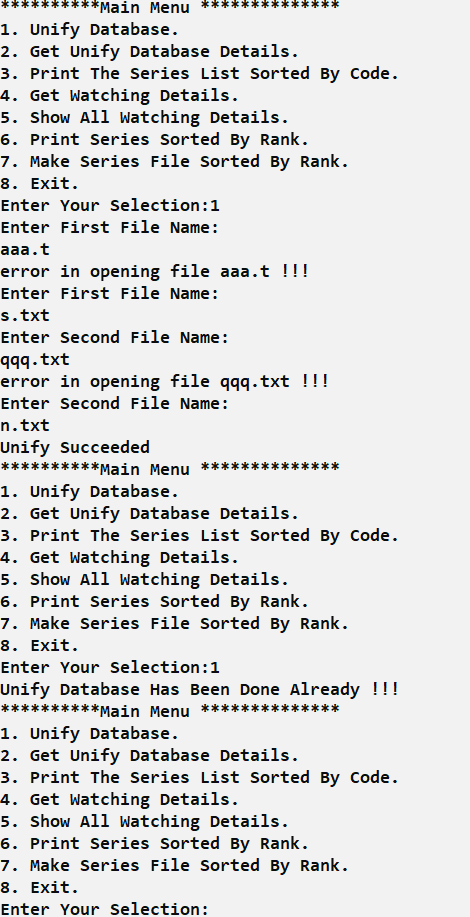
שמו של קובץ הפלט יהיה : **series.txt**

עם הבחירה באופציה זו על המשתמש להזין את שמות הקבצים של כל אחת מהחברות.

במידה ויש בעיה בפתיחת הקבצים, תוצג הודעה מתאימה והתפריט יוצג שוב.

להלן דוגמה לשימוש באופציה מספר 1:

שמות הקבצים הם, n.txt ו- s.txt .



**אופציה 2 בתפריט: Get Unify Database Details**

עם בחירת אופציה זו בתפריט ובתנאי שאופציה 1 התבצעה, התוכנית תקרא את נתוני הקובץ series.txt

הנתונים יישמרו בזיכרון לתוך מערך של מצביעים.

בנוסף, הקובץ  **watching.txt** שלתוכו ייכתבו נתוני הצפייה יאותחל שוב כקובץ ריק.

גודלו של המערך **מוגדר דינמית על פי מספר הסדרות**, כל תא במערך הוא מצביע למבנה מהצורה הבאה:

typedef struct serie {

char serieName[20];

int numSeasons;

int rank;

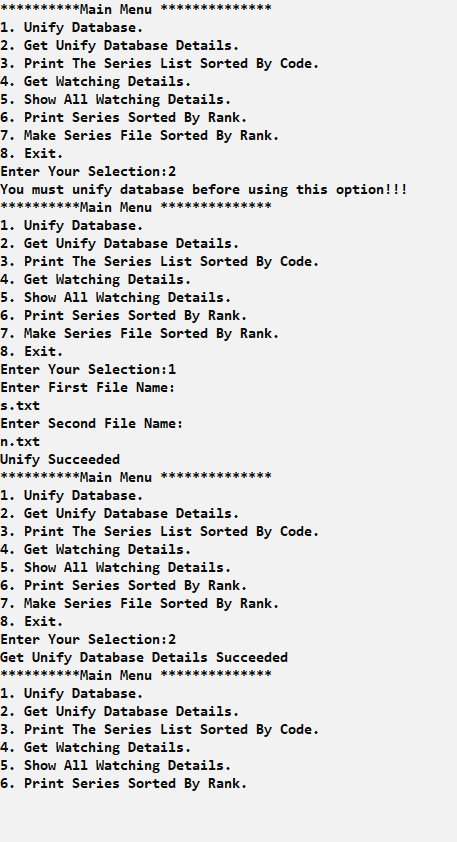
int\* watchingDetails;

}serie;

עבור כל סדרה יישמר שם הסדרה, הקוד שלה, מספר העונות וכמו כן מערך שגודלו כמספר העונות.

בתום ביצוע הפעולה תוצג הודעה מתאימה והתפריט יוצג שוב.

להלן דוגמה לשימוש באופציה מספר 2:



**אופציה 3 בתפריט:** Print The Series List Sorted By Code

עם בחירת אופציה זו בתפריט ובתנאי שאופציה 1 ו- 2 התבצעו, התוכנית תציג את נתוני הסדרות ממוינות לפי קוד סדרה בסדר עולה.

להלן דוגמה לשימוש באופציה מספר 3:



**אופציה 4 בתפריט: Get Watching Details**

עם בחירת אופציה זו בתפריט ובתנאי שאופציה 1 ו- 2 התבצעו, התוכנית תקלוט נתוני צפייה ותעדכן את מסד הנתונים בזיכרון ואת קובץ הצפיות ששמו: **watching.txt**

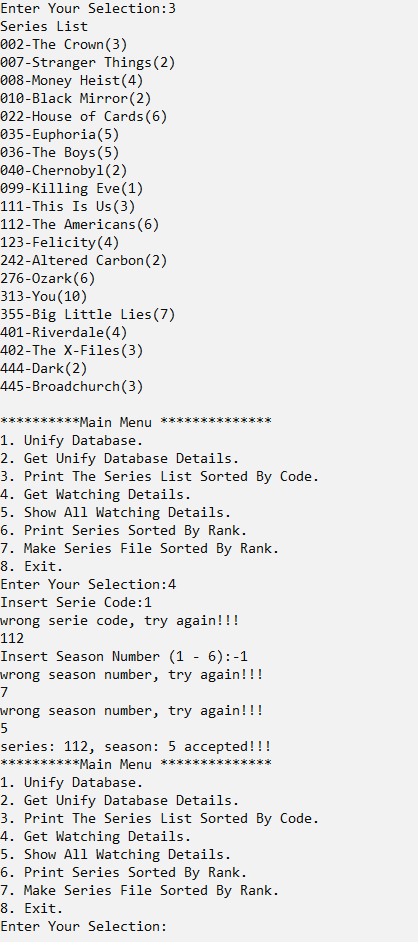
בקליטה, תחילה ייקלט קוד הסדרה ואחר כך מספר העונה,

במידה וייקלט קוד לא תקין תוצג ההודעה: wrong serie code, try again!!!

במידה וייקלט מספר עונה לא תקין תוצג ההודעה: wrong season number, try again!!!

במידה ויתקבל קלט תקין תוצג ההודעה: series: %d, season: %d accepted!!!

להלן דוגמה לשימוש באופציה מספר 4:

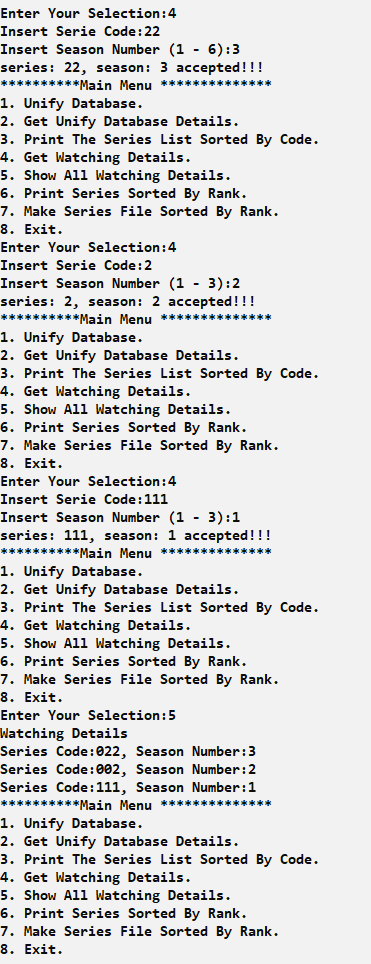


**אופציה 5 בתפריט: Show All Watching Details**

עם בחירת אופציה זו בתפריט ובתנאי שאופציה 1 ו- 2 התבצעו, התוכנית תציג את נתוני הצפייה שנקלטו.

עבור כל נתון שנקלט תוצג שורת פלט הכוללת את קוד הסדרה ואת מספר העונה.

להלן דוגמה לשימוש באופציה מספר 5:

****

**אופציה 6 בתפריט: Print Series Sorted By Rank**

עם בחירת אופציה זו בתפריט ובתנאי שאופציה 1 ו- 2 התבצעו, התוכנית תציג את נתוני הסדרות ממוינים לפי הדירוג שלהן בסדר יורד. כלומר תחילה יוצג/יוצגו הסדרות עם הדירוג הגבוה ביותר וכך הלאה עד הדירוג הנמוך ביותר. הדירוג של הסדרה הוא מספר הצפיות הכולל בסדרה (בכל העונות).

לדוגמא:

עבור הסדרה This Is Us שיש לה 3 עונות, ולפי נתוני הצפייה שנקלטו,

בעונה 1: היו 5 צפיות, בעונה 2: היו 3 צפיות ובעונה 3: היו 2 צפיות, בסה"כ 10 צפיות.

לכן דירוג הסדרה הוא 10.

תחילה יוצג הדירוג, ואחר כך יוצגו כל הסדרות בדירוג זה (ממוינות לפי קוד סדרה בסדר עולה).

לכל סדרה תוצג שורת כותרת המכילה את קוד הסדרה ושמה ולאחר מכן עבור כל עונה בסדרה יוצג מספר העונה ומספר הצפיות שהיו בעונה זו.

להלן דוגמה לשימוש באופציה מספר 6:



**אופציה 7 בתפריט: Make Series File Sorted By Rank**

עם בחירת אופציה זו בתפריט ובתנאי שאופציה 1 ו- 2 התבצעו, התוכנית תכתוב לתוך קובץ ששמו

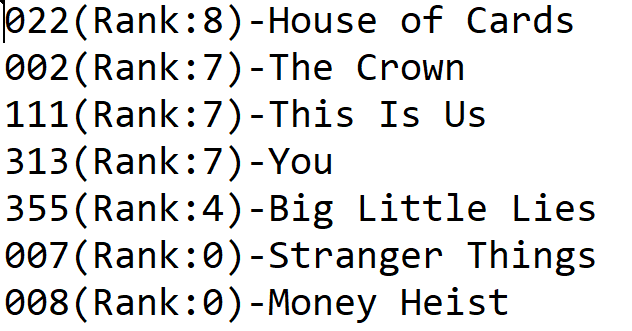
seriesRank.txt את נתוני הסדרות ממוינים לפי הדירוג שלהן בסדר יורד.

הנתונים ייכתבו לקובץ באופן הבא:

לכל סדרה תוצג שורת כותרת המכילה את קוד הסדרה ושמה והדירוג שלה,

לאחר מכן עבור כל עונה בסדרה יוצג מספר העונה ומספר הצפיות שהיו בעונה זו.

להלן דוגמא לקובץ שנוצר לאחר בחירה באופציה זו:



**אופציה 8 בתפריט: Exit**

עם בחירת אופציה זו התוכנית **תשחרר את שטחי הזיכרון שנתפסו** , תציג את ההודעה הבאה:

End Of The Program... ותסתיים.

לפניכם ההגדרות של המבנים ואב טיפוס של הפונקציות ,ניתן לשנות את חתימות הפונקציות :

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct serie {

char serieName[20];

int numSeasons;

int rank;

int\* watchingDetails;

}serie;

int showMenu();

int unifyDatabase(FILE\* , FILE\* );

serie\*\* getUnifyDatabaseDetails();

void printSeriesList(serie\*\* );

void getWatchingDetails(serie\*\* , int\* );

void printWatchingDetails();

void printSeriesByCode(serie\*\* );

void printSeriesByRank(serie\*\* , int);

void makeSeriesFileSortedByRank(serie\*\*, int);

void main() {…

**הנחות ודגשים:**

* ניתן להניח כי קבצי הסדרות קיימים והם תקינים. שם הקובץ מקסימום 19 תוים.
* יש לקרוא מכל אחד מהקבצים הנ"ל **פעם אחת בלבד**.
* יש לדאוג לניהול זיכרון תקין ולשחרור הזיכרון בסיום העבודה. חובה להשתמש בהקצאת זכרון דינמית.
* בכל מקרה של אי התאמה בין ההוראות הכתובות להתנהגות קובץ ההרצה – יש לפעול על פי קובץ ההרצה וליידע את המתרגל האחראי על אי ההתאמה.
* מצורפים קבצי קלט פלט לדוגמה. בחלק מהמקרים אין ירידת שורה בהדפסת הפלט. יש להתחיל שורה חדשה בכל הדפסה – על פי ההתנהגות של קובץ ההרצה.

**בהצלחה!**