

פרויקט מסכם תכנות מתקדם – תשפ"ב סמסטר ב'

הפרויקט מטפל במאגר מוסיקאים עבור הופעות. בהינתן קובץ טקסט (להלן: "קובץ מוסיקאים"), בכל שורה בקובץ מופיע שם של נגן המורכב ממספר מילים. לאחר מכן יופיעו כלי הנגינה עליהם המוסיקאי מנגן (לפחות אחד) כאשר אחרי כל כלי נגינה יופיע המחיר שגובה המוסיקאי להופעה אם ינגן על כלי זה. המילים שמרכיבות את השם של המוסיקאי, שמות כלי הנגינה והמחירים, מופרדות על ידי רווחים ותווים אחרים (להלן: "תווים מפרידים").

בנוסף, ישנו קובץ טקסט (להלן: "קובץ כלי נגינה") המכיל אוסף של כלי נגינה, בכל שורה רשום כלי נגינה אחד. מובטח שכל כלי נגינה שיוזכר בקובץ המוסיקאים אכן מופיע בקובץ כלי הנגינה. התכנית תקלוט מהמשתמש הופעות, כאשר להופעה יש שם, תאריך, כלי נגינה שנחוצים להופעה ובכמה מכל כלי נגינה יש צורך. כמו כן, הופעה תיתן חשיבות (1/0) לכלי נגינה, כאשר אם הכלי "חשוב" אז יעדיפו עבורו נגנים יקרים (מתוך הנחה שהם איכותיים יותר), ואחרת – יעדיפו נגנים זולים. מטרת התכנית היא לארגן הופעות על ידי שיבוץ נגנים מהמאגר להופעות. ניתן להניח כי ההופעות מתקיימות בתאריכים שונים, וכי המוסיקאים תמיד זמינים. להלן פירוט המטלות ומבני הנתונים שעליכם להגדיר.

1 בניית עץ בינארי של כלי נגינה

התכנית מקבלת כארגומנט שני פרמטרים שהם שמות של קבצי טקסט. הקובץ הראשון הוא קובץ כלי הנגינה. כאמור כל שורה בקובץ מייצגת כלי נגינה אחד. ניתן להניח שאורכה של שורה אינו עולה על 150 תווים וכי שמו של כלי הנגינה מורכב ממילה בודדת המתחילה באות גדולה. למען הסר ספק, יש לעבור על הקובץ מתחילתו עד סופו בלבד. כל צומת בעץ שומר כלי נגינה מסוים (instrument). השדה insId הוא מספר ייחודי רץ לכל כלי נגינה המתחיל מ-0 (כלומר אין שני כלי נגינה עם אותו insId). בעץ יופיע כל כלי נגינה פעם אחת. הפוינטר left בצומת נתון בעץ יצביע לתת-עץ שבו כל הערכים של instrument קטנים (לקסיקוגרפית) מהערך של instrument בצומת זה (או ל NULL במקרה שנת העץ השמאלי ריק). באותו אופן, הפוינטר right בצומת נתון יצביע לתת-עץ שבו כל המילים גדולות (לקסיקוגרפית) מהערך של instrument בצומת זה (או ל NULL במקרה שנת העץ הימני ריק).

מכל כלי נגינה שב "קובץ כלי נגינה" עליכם לבנות איבר להלן שיוכנס בהמשך לעץ חיפוש בינארי של כלי נגינה:

```
typedef struct treeNode{
    char * instrument; // שם כלי הנגינה
    unsigned short insId; // מספר ייחודי עבור כלי נגינה שבאחריותכם לתת
    struct treeNode * left;
    struct treeNode * right;
} treeNode;
```

העץ עצמו יוגדר כך:

```
typedef struct tree {
    treeNode * root;
} InstrumentTree;
```

2 המרה של שם כלי נגינה במס' המזהה שלו

כתבו את הפונקציה:

```
int findInsId(InstrumentTree tree, char *instrument);
```

הפונקציה מקבלת שם כלי נגינה ועץ ומחזירה את המספר המזהה של כלי הנגינה. אם instrument לא נמצא בעץ tree, הפונקציה מחזירה 1-.

3 בניית מאגר המוסיקאים

הארגומנט השני של התכנית הוא שם של קובץ המוסיקאים. בכל שורה בקובץ ישנו שם של נגן המורכב ממספר מילים (לפחות שתיים). ייתכן שם פרטי ושם משפחה בלבד, ייתכן גם שם אמצעי, ולפעמים גם שני שמות משפחה, ואפשר גם: "אנה ויולט דה פריז הירש". לאחר מכן יופיעו כלי הנגינה עליהם המוסיקאי מנגן (לפחות אחד), ואחרי כל כלי נגינה המחיר שגובה המוסיקאי להופעה אם ינגן על כלי זה. **ניתן להניח שאורכה של שורה אינו עולה על 150 תווים.**

עליכם לחלץ את המילים מהשורה, בהינתן תווי ההפרדה הבאים (כל התווים בין ה" ", כולל תו הרווח)

" , . ; : ? ! - \ t ' () [] { } < > ~ _ "

המילים שיש לחלץ הן המילים שמרכיבות את שמו, כלי הנגינה עליהם הוא מנגן והמחיר שהוא דורש. למשל:
Matti,, Caspi Guitar 5500 Drums (3245) Piano 7074!!

את כל המוסיקאים שרשומים בקובץ עליכם לאחסן במערך מצביעים ל-Musician בשם MusiciansGroup.

השתמשו ברשומות הבאות:

```
typedef struct
{
    unsigned short insId; // מספר מזהה של כלי נגינה
    float price; // מחיר שדורש הנגן עבור נגינה בכלי זה
} MusicianPriceInstrument;
```

```
typedef struct
{
    char **name; // מערך של השמות המרכיבים את שמו המלא
    MPIList instruments; // כלי הנגינה ברשימה מקושרת של MusicianPriceInstrument
} Musician;
```

4 בניית מאגר כלי הנגינה של המוסיקאים

עליכם לתחזק מערך של מערכים בשם MusiciansCollection של מצביעים למוסיקאים במערך MusiciansGroup. אם מוסיקאי יודע לנגן על x כלי נגינה יהיו אליו בסך הכל x הצבעות מ-MusiciansCollection. מספר איברי המערך MusiciansCollection יהיה כמספר סוגי כלי הנגינה. (שימו לב שזה מספר האיברים בעץ InstrumentTree). האיבר ה-i במערך MusiciansCollection יהיה מערך של מצביעים ל-Musician שיוודעים לנגן על כלי נגינה שמספרו המזהה i. **הניחו כי לכל כלי נגינה יש לפחות מוסיקאי אחד שמנגן עליו.**

למשל, אם ישנם 2 כלי נגינה, כינור וחצוצרה, ו-2 נגנים, אבי ויוסי, אבי יודע לנגן על שני כלי הנגינה, ויוסי רק על כינור, במערך MusiciansGroup יהיו שני איברים, עבור אבי ויוסי, ובמערך MusiciansCollection יהיו שני מערכים, אחד לכינור ואחד לחצוצרה. במערך הראשון, MusiciansCollection[0] יהיו 2 מצביעים – הראשון יצביע על אבי והשני על יוסי, ובמערך השני MusiciansCollection[1] יהיה מצביע אחד – לאבי.

5 קלט הופעה ואיתור נגנים עבורה

בהמשך, התכנית תקלוט בלולאה פרטי הופעה. עליכם להשתמש ברשומות הבאות לתיאור הופעה:

```
typedef struct
```

```
{
    int day,month,year;
    float hour;
} Date;
```

```
typedef struct
```

```
{
    int num; // כמה מאותו כלי
    int inst; // מזהה של כלי נגינה
    char importance; // האם לכלי נגינה זה יש תקציב עבור נגנים יקרים (0/1)
} ConcertInstrument;
```

```
typedef struct
```

```
{
    Date date_of_concert; // תאריך קיום ההופעה
    char *name; // שם ההופעה
    CIList instruments; // כלי הנגינה ברשימה מקושרת של ConcertInstrument
} Concert;
```

התכנית תקלוט בלולאה שורה אחרי שורה מהמקלדת הופעות. סוף הקלט הינו שורה ריקה. הקלט של הופעה יהיה בפורמט: שם ההופעה (מילה בודדת), תאריך ההופעה כ: dd mm yyyy hh:mm ואז רשימת כלי הנגינה כאשר לאחר כל כלי יש מספר הכלים מסוג זה שנחוצים להופעה והאם הכלי "חשוב" 1 או 0.

למשל :

Fields 29 01 2020 20:30 Violin 3 1 Trumpet 2 0

שם ההופעה: Fields התאריך : יום 29 חודש 1 שנה 2020 שעה 20.5 כלי נגינה: 3 כינורות – חשוב, 2 חצוצרות – לא חשוב.

עבור הופעה, יש לאחסן את הנתונים ברשומה Concert ואז לנסות למצוא נגנים מהמאגר. כל הופעה תמייין מחדש את המערך MusiciansCollection באופן הבא:

עבור כלי נגינה i אם חשיבותו 1, יבוצע מיון למערך MusiciansCollection[i] מהיקר ביותר לזול ביותר (מתוך הנחה שהיקר איכותי ומדובר בכלי נגינה חשוב להופעה). אם חשיבותו 0, המערך ימוין דווקא מהזול ליקר.

לאחר המיון, יש לעבור על המערך הממוין ולאסוף את הנגנים לפי הסדר. במידה ונגן מסוים כבר נבחר לנגן בהופעה על כלי קודם, יש לעבור לנגן הבא. במידה ואין אפשרות לארגן הופעה התכנית תדפיס למסך את ההודעה:

Could not find musicians for the concert <name>

כאשר <name> הוא שם ההופעה.

במידה והשמת נגנים להופעה הצליחה יודפסו למסך:

שם ההופעה, התאריך כ dd mm yyyy hh:mm ורשימת הנגנים: שם מלא של נגן, כלי הנגינה עליו ינגן והמחיר שייגבה. בסוף השורה יופיע המחיר הכללי של ההופעה.

הנחיות כלליות

יש לתעד את התכניות.

יש להקפיד על יעילות וחסכון בזמן ריצה וזיכרון.

יש להקפיד שגודל פונקציה לא יחרוג ממסך אחד.

יש לפנות זיכרון שהוקצה דינמית ואשר אין בו צורך יותר.

יש לשחרר את כל ההקצאות הדינמיות בסוף ריצת התכנית.

יש לבדוק אם הקצאות הזיכרון הצליחו.

יש לבדוק שפתיחת קובץ הצליחה.

יש לסגור קובץ לאחר סיום השימוש בו.

יש להשתמש ב-#define היכן שנחוץ.

אין לחרוג מה-prototype-ים שהוגדרו אך ניתן ליצור פונקציות עזר לפי הצורך.

יש להגדיר את הרשומות הנדרשות עבור הרשימות המקושרות שנדרשות בפרויקט.

ניתן להוסיף שדות לרשומות אשר נתונות בפרויקט אולם אין לגרוע מהן.

ניתן להגדיר טיפוסים חדשים.

ניתן להניח כי כל קלט מהמשתמש הינו תקין וכל הטקסטים בקבצים תקינים.