

## מבוא לתכנות מערכות 10010 סמסטר א' – תשפ"ד

### תרגיל בית מס' 3

#### נושא התרגיל:

#### קבצים, void\* מצביעים לפונקציות ורשימות מקושרות

תרגיל הבית יוגש בזוגות על פי רישום מוקדם

יש להגיש את התרגיל בשני מקומות:

1. בסביבת casain - מערכת לבדיקה אוטומטית, הקישור ב Moodle

2. תיבת הגשה מתאימה של מטלה ב Moodle עבור בדיקה ידנית

#### הנחיות להגשה בקזאין:

- רק חבר קבוצה אחת יעלה את העבודה לקזאין!!!
- יש לעלות למערכת את כל הקבצים למעט קובץ ה main.c .
- בקזאין הוגדר ה main שלה, יש להשתמש בו! ולא לשנותו למעט פונקציה אחת שיש לממש!! ובנוסף לעלות את קבצי הקוד.

#### להגשה עבור הבדיקה הידנית:

- התרגיל ייבדק בסביבת Window - Visual Studio 2022
- הקוד חייב לעבור קומפילציה, קוד שאינו מתקמפל לא ייבדק.
- יש להגיש קוד ללא הערות קומפילציה, warnings, קוד בו יהיו הערות יגורר הורדה של 10 נקודות.

- העבודה תעלה למודל ע"י חבר קבוצה **אחד**
- שם הקובץ יהיה כשמות המגישים (שם פרטי ומשפחה בלבד)
- יחד עם כל הקבצים ב zip יש להוסיף קובץ **readme.txt** בו פירוט כל חברי הקבוצה בעברית
- יש להגיש את קבצי הקוד וקבצי הפרויקט sln ו vcxproj , אנשי המק יכולים להגיש רק את קבצי הקוד .
- אין להגיש קבצי obz והרצה (יגורר הורדה של 5 נקודות).

#### הוראות כלליות:

1. יש להקפיד על כללי הנדסת התוכנה:

1. פונקציה לא אמורה להיות ארוכה מ 25 שורות. אם יש קטע קוד שניתן לתת לו כותרת מה הוא מבצע יש לכתוב אותו בפונקציה נפרדת.
2. יש לחלק את הקוד לתת פונקציות מתאימות. אין לשכפל קוד. קוד זהה אמור להיות בפונקציה מתאימה.
3. מבנה התכנית (הזחות) ותיעוד במידת הצורך .
4. חובה להשתמש בקבועים במקומות המתאימים .
5. יש להשתמש בפונקציות קצרות, כלליות, קריאות ושימושיות.
6. יש להקפיד על בדיקת תקינות קלט, אפשר להניח שאם ביקשו מהמשתמש מספר הוא הכניס מספר אך יתכן ולא בטווח הנכון.
7. הפלט צריך להיות כפי שניתן בתרגיל במידה וניתן.
8. קוד קצר, לא מסורבל ויעיל הן מבחינת כתיבתו והן מבחינת ריצת התוכנית.

### פירוט תרגיל בית 3

בתרגיל זה נמשיך לממש את מערכת ניהול נתוני הטיסות מתרגיל 2.

### חובה להמשיך מפתרון תרגיל בית 2 שפרסמתי במודל

#### רשימות מקושרות :

1. נתון קוד של רשימה מקושרת חד כיוונית השומר נתון int. עדכן אותה כך שתהיה כללית, כלומר ה Node שומר ערך מסוג void\*. (חובה להשתמש בקוד המצורף!) הפונקציות הנדרשות:

1. איתחול רשימה
2. הוספת איבר לרשימה
3. מחיקת איבר מהרשימה – צור פונקציה כללית שתומכת במצב בו יש לשחרר את הערך ב Node ובמצב שאין צורך לשחרר את ערך זה.
4. שחרור כל הרשימה
5. הדפסת כל הרשימה

2. נשנה את המבנה של **רשות שדות התעופה**, כך שיחזיק את שדות התעופה כרשימה מקושרת השומרת \*Airport. יש לדאוג ששדות התעופה יהיו מסודרים לפי קוד שדה התעופה. יש למחוק

את המערך הקיים ואת המשתנה airportsCount!!!

אין למיין את הרשימה בשום שלב אלא להכניס את האיבר למקום הנכון עבורו, אין להשתמש במבנה עזר מכל סוג שהוא!

#### חברת התעופה

חברת התעופה מאפשרת למיין את מערך הטיסות שלה לפי השדות הבאים:

- קוד מקור
- קוד יעד
- תאריך

על מנת שנדע לפי איזה שדה הטיסה ממוינת יש להוסיף תכונה מתאימה לחברת התעופה (enum שיכיל גם את האופציה שהמערך אינו ממוין). בהתחלה המערך לא ממוין כלל.

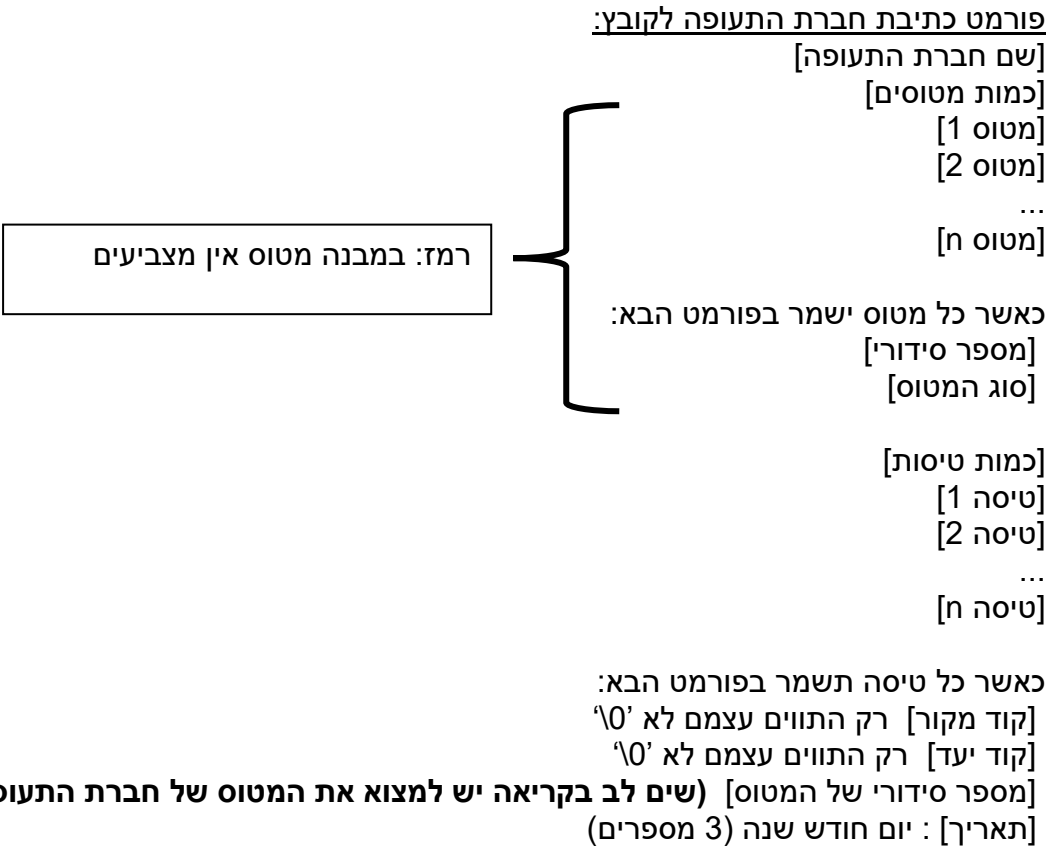
יש להשתמש בפונקציה **qsort** מ-stdlib על מנת לבצע את המיון.

יש להוסיף יכולת חיפוש טיסה במערך הממוין ע"פ הפרמטר בו הם ממוינים, יש להשתמש בחיפוש בינארי פונקציית הספרייה: **bsearch**, במידה והמערך אינו ממוין יש להודיע למשתמש שלא ניתן לבצע חיפוש.

**קבצי binary:**

את נתוני חברת התעופה נשמור לקובץ binary הקובץ יישמר תחת השם "airline.bin" כל מחרוזת תשמר בפורמט הבא:

- מספר שמייצג את מספר התווים במחרוזת כולל '0'
- המחרוזת עצמה



[Type here]

## מצביעים לפונקציה

יש להכין פונקציה גנרית המטפלת במערך, `generalArrayFunction`. הפונקציה תקבל מערך, מספר אברים במערך, גודל כל איבר בבתיים וכן מצביע לפונקציה

```
void f(void* element);
```

יש להפעיל את הפונקציה הזו בחברת התעופה במקרים הבאים:

1. הדפסת מערך טיסות.
2. הדפסת מערך המטוסים
3. שחרור מערך המטוסים
4. שחרור מערך טיסות

## שינויים ב main

### הפונקציות שיש לממש עבור ה main בקזאין ובכלל:

```
int initManager(AirportManager* pManager, const char* fileName)
```

- פונקציה המאתחלת את רשות שדות התעופה מהקובץ במידה והצליחה
- במידה והיא לא מצליחה לאתחל מהקובץ היא מאתחלת רשימה ריקה
- הפונקציה מחזירה:
  - 0 לשגיאה
  - 1 אם הקריאה מהקובץ הצליחה
  - 2 אם הקריאה מהקובץ לא הצליחה אך איתחול הרשימה תקין (עובדים מהמשתמש ללא קבצים)

```
int saveManagerToFile(const AirportManager* pManager, const char* fileName)
```

- פונקציה השומרת את פרטי רשות שדות התעופה לקובץ במידה והצליחה מחזירה 1 אחרת 0

```
int initAirlineFromFile(Airline* pComp, AirportManager* pManager, const char* fileName)
```

- פונקציה המאתחלת את חברת התעופה מהקובץ במידה והצליחה מחזירה 1 אחרת 0

```
int saveAirlineToFile(const Airline* pComp, const char* fileName)
```

- פונקציה השומרת את פרטי חברת התעופה לקובץ במידה והצליחה מחזירה 1 אחרת 0

```
int initManagerAndAirline(AirportManager* pManager, Airline* pCompany)
```

- הפונקציה מאתחלת את רשות שדות התעופה. במידה והייתה שגיאה יוצאת
- במידה והאתחול מהקובץ היה מוצלח מאתחלת את חברת התעופה מקובץ.
- במידה והאתחול מהקובץ לא היה מוצלח מאתחלת את חברת התעופה מהמשתמש כמו בתרגיל 2.

[Type here]

בתחילת התוכנית נאתחל את המערכת ע"י טעינת נתוני רשות שדות התעופה וחברת התעופה מהקבצים המתאימים. ביציאה מהתוכנית יש לשמור את נתוני חברת התעופה ורשות שדות התעופה לקבצים המתאימים.

### נוסף מספר אפשרויות לתפריט הראשי:

1. מיון טיסות לפי תכונה נתונה בסדר עולה (תפריט פנימי ע"פ ערכי ה-enum).  
הפונקציה

**void      sortFlight(Airline\* pComp)**

2. חיפוש טיסה בחברת התעופה לפי התכונה בה המערך ממוין

**void      findFlight(const Airline\* pComp)**

**בהצלחה!**