

תרגיל בית 2 – פיתוח מודולרי ורשימות מקושרות

מועד ההגשה: יום ב', 12/12/2016 בשעה 23:59
האחראי על התרגיל: הגר פורת
hagarp@campus

מטרה: שימוש ברשימות מקושרות על מנת ליצור מערכת לניהול שדה תעופה.

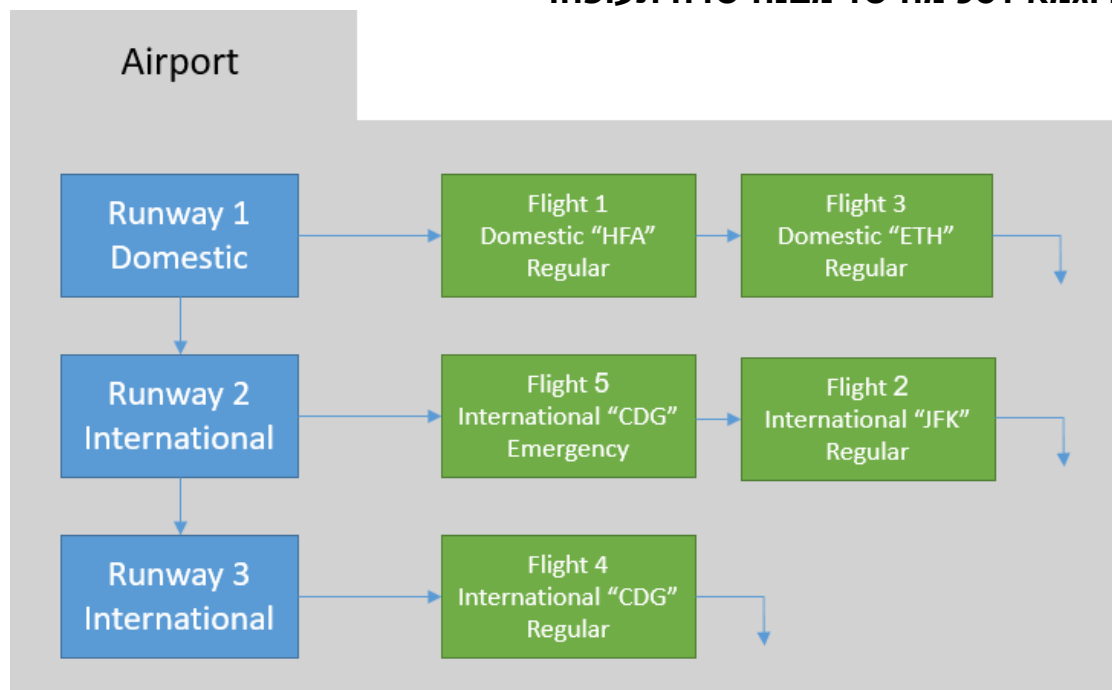
רקע:

בכדי לייעל את שדה התעופה, הוחלט במרכזי הבקרה לעבור למערכת חדשנית ולא להסתמך יותר על המערכות הישנות.
בשדה התעופה יש מסלולי המראה משני סוגים, עבור טיסות פנימיות ועבור טיסות בינלאומיות. כל מסלול מוגדר ע"י מספר מסלול (לא ייתכן כי קיימים שני מסלולים בעלי אותו מספר בשדה התעופה).
בנוסף, לכל מסלול מוגדרת רשימה של טיסות המחכות להמראה.
טיסה מוגדרת ע"י מספר טיסה (לא ייתכן כי קיימות שתי טיסות בשדה התעופה עם מספר טיסה זהה). כמו כן, לכל טיסה מצוין האם זו טיסת חוץ או פנים, יעד הטיסה והאם זאת היא טיסת חירום.

נשתמש בשלושה מבני נתונים:

1. טיסה – מבנה נתונים אשר יכיל את מספר הטיסה, האם זו טיסת פנים או חוץ, יעד הטיסה והאם זו טיסת חירום.
2. מסלול המראה – מבנה נתונים אשר יכיל את מספר המסלול, האם זה מסלול המיועד לטיסות פנים או חוץ ואת רשימת הטיסות הממתנות להמראה.
3. שדה תעופה – מבנה נתונים אשר יממש את שדה התעופה עצמו ויכיל את המסלולים והטיסות.

דוגמא לסכימה של מבנה שדה תעופה:



איך נקבע מיקום הטיסה:

כאשר מתווספת טיסה חדשה לשדה התעופה, נכניס את הטיסה לסוף הרשימה של מסלול המיועד לאותו סוג של טיסה. כלומר, טיסת פנים תיכנס למסלול לטיסות פנים וכנ"ל לגבי טיסות חוץ. במידה ויש כמה מסלולים מאותו הסוג אז הטיסה תיכנס למסלול המכיל נכון לאותו הזמן פחות טיסות. אם יש את אותה כמות טיסות בכל המסלולים אז נבחר את המסלול שמספרו הכי נמוך. טיסת חירום תיכנס לתחילת הרשימה, אך לאחר טיסות החירום האחרות שממתינות באותו המסלול שנכנסו קודם לכן, אם קיימות כאלה.

דוגמא בסכימה:

טיסות הפנים נכנסו למסלול 1 המיועד לטיסות פנים. טיסה 1 התווספה קודם לשדה התעופה ולכן היא הטיסה הראשונה ברשימה. לאחר מכן נוספו טיסות 2,3,4,5 לפי הסדר. טיסה 2 בינלאומית נכנסה למסלול בינלאומי מספר 2 מכיוון שתחילה שני המסלולים הבינלאומיים היו ריקים ומסלול מספר 2 הוא בעל המספר הנמוך ולכן הטיסה נכנסה לשם. טיסה 3 היא טיסת פנים ולכן נכנסה למסלול 1 אחרי טיסה מספר 1. טיסה 4 היא טיסה בינלאומית וכעת מסלול 3 הוא בעל 0 טיסות ולעומתו מסלול 1 עם טיסה אחת ברשימה. לכן טיסה 4 נכנסה למסלול 3. טיסה 5 נכנסה למסלול 2 מכיוון שכל המסלולים הבינלאומיים מכילים טיסה אחת והוא המסלול עם המספר הנמוך יותר. טיסה 5 היא טיסת חירום ולכן נכנסה לתחילת הרשימה.

שימו לב: דוגמאות נוספות מופיעות בקבצי הבדיקות (קבצי קלט פלט) המצורפים.

חלק 1 – ממשק טיסה:

תחילה נממש ממשק עבור טיסה. הממשק יופיע בקובץ `flight.h` והמימוש בקובץ `flight.c`.

עבור כל טיסה נשמור:

1. מספר טיסה – מספר הטיסה ייחודי בשדה התעופה. מספר הטיסה הינו מספר טבעי בין הערך 1 לערך `MAX_ID` המוגדר בקובץ `ex2.h`.
2. סוג הטיסה – על פי הטיפוס `FlightType` המוגדר בקובץ `ex2.h` ויכול לקבל את הערכים `{DOMESTIC, INTERNATIONAL}`.
3. יעד הטיסה – מחרוזת בעלת 3 אותיות גדולות באנגלית לדוגמא: `TLV,JFK`.
4. חירום – האם זו טיסת חירום על פי הטיפוס `BOOL` המוגדר בקובץ `ex2.h` ויכול לקבל את האחרים `{TRUE,FALSE}`.

על ממשק הטיסה לתמוך בפעולות הבאות:

createFlight – הפונקציה מקבלת מספר טיסה, סוג, יעד והאם זו טיסת חירום. באמצעות פרמטרים אלו יוצרת טיסה חדשה. יש להשתמש בהקצאה דינמית.

הפונקציה תחזיר מצביע לטיסה החדשה שנוצרה.

הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל
מבוא למערכות תוכנה

במידה והפונקציה נכשלה יוחזר NULL (סיבות לכישלון עלולות להיות פרמטרים לא חוקיים, הקצאת זיכרון שנכשלה).

destroyFlight – הפונקציה מקבלת מצביע לטיסה ומשחררת את הטיסה.

printFlight – הפונקציה מקבלת מצביע לטיסה ומדפיסה את פרטי הטיסה. במידה והמצביע שהתקבל הוא NULL הפונקציה תחזיר FAILURE ואחרת SUCCESS (על פי הטיפוס Results המוגדר בקובץ ex2.h).
אופן ההדפסה יהיה:

Flight 1 D TLV R

עבור טיסת פנים מספר 1 ליעד TLV שהיא טיסה רגילה

Flight 10 I CDG E

עבור טיסת חוץ מספר 10 ליעד CDG שהיא טיסת חירום

חלק 2 – מנשק מסלול המראה:

המנשק שנממש יופיע בקובץ runway.h והמימוש בקובץ runway.c.
עבור כל מסלול נשמור:

1. מספר המסלול – מספר המסלול ייחודי בשדה התעופה. מספר המסלול הינו מספר טבעי בין הערך 1 לערך MAX_ID המוגדר בקובץ ex2.h.
2. סוג המסלול – על פי הטיפוס FlightType המוגדר בקובץ ex2.h ויכול לקבל את הערכים {DOMESTIC, INTERNATIONAL}.
3. רשימת הטיסות של המסלול – תמומש על ידי רשימה מקושרת חד-כיוונית שתכיל את הטיסות (כפי שניתן לראות בסכימה).

על מנשק זה לתמוך בפעולות הבאות:

createRunway – הפונקציה מקבלת מספר המסלול ואת סוג המסלול ויוצרת מסלול חדש עם רשימת טיסות ממתונות ריקה. יש להשתמש בהקצאה דינמית עבור המסלול.

הפונקציה תחזיר מצביע למסלול החדש שנוצר.

במידה והפונקציה נכשלה יוחזר NULL (סיבות לכישלון עלולות להיות פרמטרים לא חוקיים, הקצאת זיכרון שנכשלה).

destroyRunway – הפונקציה מקבלת מצביע למסלול ומוחקת את המסלול ואת רשימת הטיסות הממתונות במסלול זה.

isFlightExists – פונקציה זו תקבל מצביע למסלול ומספר טיסה ותחזיר TRUE אם הטיסה נמצאת ברשימת הטיסות הממתונות במסלול, אחרת FALSE. הפונקציה תחזיר FALSE.
במידה והפונקציה נכשלה יוחזר FALSE (סיבות לכישלון עלולות להיות פרמטרים לא חוקיים).

getFlightNum – הפונקציה תקבל מצביע למסלול ותחזיר את מספר הטיסות הממתונות במסלול. במידה והפונקציה נכשלה יוחזר 1- (סיבות לכישלון עלולות להיות פרמטרים לא חוקיים).

addFlight – הפונקציה תקבל מצביע למסלול ומצביע לטיסה, ותוסיף את הטיסה לרשימת הטיסות הממתונות במסלול.
ההכנסה לרשימת הטיסות תבוצע על פי החוקים שנקבעו בתחילת התרגיל.
אם קיימת טיסה בעלת אותו מספר ברשימה הפונקציה תיכשל.
אם סוג הטיסה לא מתאים לסוג המסלול הפונקציה תיכשל.
על מבנה הנתונים runway ליצור עותק של flight (רמז: קיימת פונקציה לכך במנשק flight).
במידה והפונקציה נכשלה יוחזר FAILURE, אחרת יוחזר SUCCESS
(סיבות נוספות לכישלון עלולות להיות פרמטרים לא חוקיים, הקצאות זיכרון שנכשלו).

removeFlight – הפונקציה תקבל מצביע למסלול ומספר טיסה ותסיר את הטיסה מהרשימה. אם לא נמצא טיסה בעלת המספר שניתן כפרמטר, אין לבצע שינויים ברשימה ויש להחזיר FAILURE.
במקרה שההסרה בוצעה בהצלחה יש להחזיר SUCCESS.
במידה והפונקציה נכשלה יוחזר FAILURE (סיבות נוספות לכישלון עלולות להיות פרמטרים לא חוקיים).

depart – הפונקציה תקבל מצביע למסלול ותסיר את הטיסה הראשונה מרשימת הטיסות הממתונות במסלול זה. במידה והמצביע שהתקבל הוא NULL הפונקציה תחזיר FAILURE, אחרת SUCCESS.

printRunway – הפונקציה תקבל מצביע למסלול ותדפיס את מספרו, את סוגו ואת פרטי הטיסות ברשימתו על פי הסדר.
במידה והמצביע שהתקבל הוא NULL הפונקציה תחזיר FAILURE, אחרת SUCCESS.
אופן ההדפסה יהיה:

```
Runway 2 international
2 flights are waiting:
Flight 5 I CDG E
Flight 2 I JFK R
```

חלק 3 – מנשק שדה תעופה:

כעת נממש את שדה התעופה עצמו על ידי רשימה מקושרת שתכיל את המסלולים הקיימים בשדה.
מנשק זה יופיע בקובץ airport.h והמימוש יופיע בקובץ airport.c.

על שדה התעופה לתמוך בפעולות הבאות:

addRunway – הפונקציה תקבל מספר של מסלול וסוג מסלול

הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל

מבוא למערכות תוכנה

{DOMESTIC, INTERNATIONAL}, תיצור מסלול חדש לפי הפרמטרים הללו ותכניס את המסלול **לסוף** רשימת המסלול. לא ייתכן כי בשדה קיימים שני מסלולים עם אותו מספר. אם מסיבה כלשהי התרחשה שגיאה בפונקציה יש להחזיר FAILURE, אחרת יש להחזיר SUCCESS.

removeRunway – הפונקציה תקבל מספר של מסלול ותסיר את המסלול מהשדה. במקרה זה, יש להסיר את כל הטיסות אשר ממתינות במסלול זה מן השדה.

אם לא קיים מסלול עם מספר כזה או התרחשה שגיאה בפונקציה יש להחזיר FAILURE, אחרת יש להחזיר SUCCESS.

addFlightToAirport – הפונקציה תקבל פרטים של טיסה: מספר טיסה, סוג טיסה, יעד ומשתנה מסוג BOOL שיגדיר האם זו טיסת חירום. הפונקציה תכניס את הטיסה למסלול המתאים (כמו שהוגדר בהתחלה). אם כבר קיימת טיסה בעלת מספר טיסה כזה **בשדה**, או הפרמטרים לא תקינים או שלא קיים מסלול מתאים או שהתרחשה שגיאה כלשהי בפונקציה יש להחזיר FAILURE, אחרת אם הטיסה הוכנסה לרשימה המתאימה יש להחזיר SUCCESS.

departFromRunway – הפונקציה תקבל מספר של מסלול המראה ואם קיים מסלול כזה אז היא תסיר את הטיסה הראשונה המופיעה ברשימה שלו. אם לא קיים מסלול בעל המספר הנתון או שהתרחשה שגיאה כלשהי בפונקציה יש להחזיר FAILURE, אחרת אם הטיסה הוסרה בהצלחה יש להחזיר SUCCESS.

stormAlert – הפונקציה תקבל שם של יעד. אם היעד לא חוקי (כמוגדר במנשק טיסה) יש להחזיר FAILURE. הפונקציה תעבור על כל המסלולים ותחפש טיסות שיעדן הוא היעד שהוגדר. במידה ונמצאו טיסות כאלה עלינו להוציא אותן מרשימת הטיסות של המסלול ולהחזירן ככה שיתווספו לסוף הרשימה. אם זוהי טיסת חירום אז היא תיכנס לרשימה אחרי כל טיסות החירום האחרות. אם קיימות כמה טיסות ליעד זה ברשימה אז הסדר הפנימי שלהן ישמר.

לדוגמא, אם זה היה המצב של השדה לפני:

Airport status:
Runway 1 international
3 flights are waiting:
Flight 1 | CDG E
Flight 2 | JFK E
Flight 3 | CDG R
Runway 2 international
3 flights are waiting:

Flight 4 I CDG R
Flight 5 I CDG R
Flight 6 I JFK R

אז אחרי שהתקבלה התראה על סופה ב- CDG המצב של השדה יהיה:

Airport status:
Runway 1 international
3 flights are waiting:
Flight 2 I JFK E
Flight 1 I CDG E
Flight 3 I CDG R
Runway 2 international
3 flights are waiting:
Flight 6 I JFK R
Flight 4 I CDG R
Flight 5 I CDG R

printAirport – הפונקציה תדפיס את שדה התעופה באופן הבא:

Airport status:
Runway 1 domestic
3 flights are waiting:
Flight 1 D HFA R
Flight 3 D ETH R
Runway 2 international
2 flights are waiting:
Flight 5 I CDG E
Flight 2 I JFK R
Runway 3 international
1 flights are waiting:
Flight 4 I CDG R

destroyAirport – הפונקציה תמחק את שדה התעופה, המסלולים והטיסות.

שימו לב: ניתן להוסיף לכל המנשקים פונקציות נוספות במידת הצורך.

חלק 4 – פונקציית main:

עליכם לכתוב קובץ main.c שיכיל פונקציית main, יבצע ניתוח של הקלט ויענה על הדרישות הבאות: התוכנית תקלוט שורה מהקלט הסטנדרטי ותבצע את הפעולה המתאימה. הניחו כי אורך שורת הקלט המקסימלי הינו 256 תווים.

כל שורת קלט תהיה אחת מהצורות הבאות:

1. הכנסת מסלול חדש לשדה התעופה

Insert <runwayNumber> <runwayType>

כאשר runwayType יכול להיות I (עבור טיסות בינלאומיות) או D (עבור טיסות פנים).

2. הסרת מסלול משדה התעופה

Remove <runwayNumber>

3. הוספת טיסה לשדה התעופה

Add <flightNumber> <flightType> <destination> <emergency>

כאשר flightType יכול להיות I (עבור טיסות בינלאומיות) או D (עבור טיסות פנים) ו- emergency יכול להיות E (עבור טיסות חירום) או R (עבור טיסות רגילות).

4. המראה ממסלול

Depart <runwayNumber >

5. התראה על סופה ביעד

Storm <destination>

6. הדפסת שדה התעופה

Print

7. יציאה מהתכנית (יש לשחרר את מבנה שדה התעופה ולסיים את התכנית)

Exit

לא ניתן להניח שהקלט תקין, כלומר:

אם יש שגיאה בקלט, יש להדפיס הודעה מתאימה ל-Standard Error ולעבור לשורת הקלט הבאה.

את ההדפסה ל-Standard Error יש לבצע ע"י:

```
fprintf(stderr, <errorMessage>);
```

<errorMessage> יפורט בהמשך עבור כל סוג שגיאה.

השגיאות בהן יש לטפל הן:

- חוסר בפרמטרים – יש להדפיס את שם הפקודה ואחריה
"not enough parameters". לדוגמא עבור פקודת Add נדפיס:
"Add failed: not enough parameters."
עבור פקודת Insert נדפיס:
"Insert failed: not enough parameters."
- פקודה לא מוכרת – במקרה זה יש להדפיס:
"Command not found."
- עבור כישלון של פקודות ועבור פרמטרים לא תקינים, למשל ערכים לא נומריים עבור פרמטר נומרי, יש להדפיס הודעה בסגנון:
"Add execution failed."

במקרה שיש עודף פרמטרים, יש להתעלם מפרמטרים עודפים ולא להתייחס לכך כשגיאה.

התוכנית תשחרר את הרשומות הנותרות ותסתיים כאשר נגמר הקלט (גם אם לא התקבלה פקודת Exit).

הטכניון - מכון טכנולוגי לישראל

הפקולטה להנדסת חשמל
מבוא למערכות תוכנה

לעזרתכם דוגמה של תוכנית המבצעת ניתוח של הקלט תוך שימוש בפונקציות
atoi, strtok, fgets

```
int main() {
    char szLine[MAX_LINE_SIZE];
    char* delimiters = "\\t\\n";
    char* pszName;
    char* pszYear;
    int year;

    while (fgets(szLine, MAX_LINE_SIZE, stdin)) {
        printf("> %s", szLine);
        pszName = strtok(szLine, delimiters);
        pszYear = strtok(NULL, delimiters);
        if (pszName == NULL || pszYear == NULL)
        {
            fprintf(stderr, "Error: Not enough parameters\\n");
        }
        year = atoi(pszYear);
        if (year == 0)
        {
            fprintf(stderr, "Error: Invalid year\\n");
        }
        printf("%s is %d years old.\\n", pszName, CURR_YEAR -
year);
    }
    return 0;
}
```

תוכנית זו מצפה לקבל בכל שורת קלט שם ושנת לידה ומדפיסה את הגיל.

המלצות: פונקציות C שימושיות לתרגיל, strtok, strcmp, strlen, strcpy, malloc, free, fgets.

בדיקות: לעזרתכם קבצי בדיקה in_#.txt קובץ פלט מצופה out_#.txt וקובץ שגיאות בסיס err_#.txt. (שימו לב קובץ הפלט לא מכיל את ה- stderr ולכן לא מופיעות בו הודעות השגיאה).

תרגילון :bash

רשום script קצר בשם starts_with_letter המקבל כפרמטר שם של תיקייה הנמצאת בתיקייה הנוכחית ואות באנגלית ומדפיס לפלט הסטנדרטי את מספר הקבצים בתיקייה הנתונה (או בתתי תיקיות שלה) המתחילים באות זו. למשל, אם האות היא z אז קבצים המתחילים באות z או Z יחשבו. אם לא התקבלו מספיק פרמטרים יודפס "Not enough parameters". אם התיקייה לא קיימת יודפס "Directory does not exist".

הטכניון - מכון טכנולוגי לישראל
הפקולטה להנדסת חשמל
מבוא למערכות תוכנה

הנחיות הגשה:

1. קבצי קוד חלקיים, וכן קבצי קלט ופלט לדוגמה, נמצאים בתיקייה:
~eesoft/hmw/hmw2
לפני תחילת העבודה, הורידו את הקבצים לחשבונכם באמצעות הפקודה:
cp ~eesoft/hmw/hmw2/* .
2. עברו היטב על הוראות ההגשה של תרגילי הבית המופיעים באתר טרם ההגשה! ודאו כי התכנית שלכם עומדת בדרישות הבאות:
 - התכנית קריאה וברורה
 - התכנית מתועדת היטב לפי דרישות התייעוד המופיעות באתר
 - התכנית מתקמפלת ללא שגיאות וללא warnings כלל
 - התכנית רצה ללא דליפות זיכרון וגישות לא חוקיות לזיכרון כלל (בדיקה באמצעות valgrind)
 - התכנית נותנת פלט **זהה לחלוטין** לפלט הצפוי על כל קבצי הקלט שסופקו (בדיקה באמצעות פקודת diff על קבצי הפלט)
 - קובץ ה-makefile יוצר קובץ הרצה בשם הנדרש
3. יש להגיש קובץ tar יחיד המכיל את **כל הקבצים** שאתם נדרשים להגיש **ואותם בלבד** – ללא תתי-תיקיות. ודאו כי לא שכחתם את קובץ readme המכיל את פרטי הסטודנטים, וכן את ה-makefile במידה ונדרשתם.
4. שאלות בנוגע לתרגיל יש להפנות לפורום התרגיל ב-moodle בלבד – ניתן לשלוח שאלות במייל **למתרגל האחראי על התרגיל בלבד**, ורק במידה והשאלה מכילה פתרון חלקי.
5. סיכום מפרט התרגיל:

סעיף			תיאור
נושא התרגיל			C, רשימות מקושרות
תאריך ההגשה			יום ב', 12/12/2016 בשעה 23:59
המתרגל האחראי על התרגיל			הגר פורת hagarp@campus
תיקייה המכילה קבצים לשימוש הסטודנטים			~eesoft/hmw/hmw2
קבצי הקוד הנתונים			ex2.h parse_example.c
קבצי הקלט והפלט הנתונים			in_1.txt out_1.txt err_1.txt in_2.txt out_2.txt err_2.txt
הקבצים שיש להגיש			readme makefile ex2.h flight.c flight.h runway.c runway.h airport.c airport.h main.c starts_with_letter
שם תכנית ההרצה הדרושה (הנוצרת ע"י makefile)			airport
דגשים מיוחדים			...

בהצלחה!