

מבוא למדעי המחשב

מטלה 6

נושא: הקצאות דינמיות ומבנים

מתרגלים אחראים: ענבל אברהם ואביב שוקרון

הוראות כלליות

המטלה כוללת 4 קבצים:

mainTrain.c – קובץ ה-main שלכם, הוא מריץ בדיקות על הפונקציות שתממשו.
Apartment.h – קובץ הגדרות עבור המבנה Apartment והפונקציות הנחוצות לו.
Floor.h – קובץ הגדרות עבור המבנה Floor והפונקציות הנחוצות לו.
Building.h – קובץ הגדרות עבור המבנה Building והפונקציות הנחוצות לו.

במטלה זו תתבקשו לממש מבנה נתונים (struct) ע"י מימוש של 13 פונקציות. יש להוריד את המטלה ממערכת ההגשות האוטומטית ולהשתמש בקבצי ה-H המקוריים על מנת שלא יתגלו שגיאות קומפילציה. למשל, שלא רשמתם שמות של פונקציות בצורה נכונה.

עליכם לייצר 3 קבצי C (יש לשים לב לאותיות גדולות בשם הקובץ אחרת תהיה שגיאה במערכת):

Apartment.c – בו תממשו את הפונקציות שיש ב-Apartment.h

Floor.c – בו תממשו את הפונקציות שיש ב-Floor.h

Building.c – בו תממשו את הפונקציות שיש ב-Building.h

שימו לב!

בכל שאלה אתם רשאים להשתמש בקוד שמימשתם מסעיפים קודמים. למשל אם הפונקציה משאלה מספר 2 יכולה להשתמש בפונקציה שמימשתם בשאלה 1 אתם רשאים לקרוא לפונקציה מ-1 בפונקציה של 2.

לאחר שתממשו, תוכלו לקמפל ולהריץ את התכנית עם ה-main המוכן או לשנות את ה-main כראות עיניכם ולבצע עוד בדיקות משלכם.

חשוב!

יש לוודא הרצה של התכנית בסביבה שלכם לפני הגשה למערכת האוטומטית. בצורה זו תוכלו ללמוד ולהתפתח בצורה הטובה ביותר.

בסיום המטלה הנכם נדרשים להגיש את **3 קבצי ה-C שיצרתם - Apartment.c, Floor.c ו-Building.c** ובהם המימושים של השאלות.

לא לשכוח לרשום הערות לאורך הקוד (כל 2-3 שורות).

כמו כן הנכם מתבקשים לרשום הערה ארוכה בתחילת הקובץ Building.c הכוללת שם, תזז ותאריך. כמו כן הנכם מתבקשים לרשום לפני כל פונקציה הערה ובה מה הפונקציה עושה.

ברצוננו לממש מאגר נתונים בסיסי עבור בניין מגורים.
לשם כך יש לנו את ההגדרות הבאות:

```
typedef struct Apartment
{
    char* family_name;           //the name of the family
    int number;                  //the apartment number
    float number_of_rooms;       //how many rooms are in the apartment
    float area;                  //the apartment area in meters
} Apartment;

typedef struct Floor
{
    Apartment** apartments;      //a pointer to an array of pointers to apartments
    int num_of_apartments;        //number of apartments in this floor
    int floor_number;             //the number of the floor
    char* floor_title;           //the title of the floor (Normal, Penthouse, Garden)
} Floor;

typedef struct Building
{
    char* street_name;           //the name of the street
    int street_number;           //the number of the building in this street
    int num_of_floors;           //number of floors
    Floor** floors;              //a pointer to an array of pointers to floors
} Building;
```

על מנת להחזיק את מאגר הנתונים הנ"ל ברצוננו לממש את הפונקציות הבאות.

1.

Apartment* CreateNewApartment (const char* _fname, int _number, float _rooms, float _area);
הפונקציה מקבלת כפרמטר שם, מספר דירה, מספר חדרים ושטח ומחזירה דירה חדשה (שמוקצה דינמית) כאשר השדות של הדירה מאותחלים בהתאמה בפרמטרים שהתקבלו מהפונקציה. לא לשכוח לבצע הקצאות לכל המשתנים שזקוקים לכך, על מנת למשל ששם המשפחה יועתק ולא יצביע לאותו מקום בזיכרון כמו המקורי. במידה ואחת ההקצאות בפונקציה לא מצליחה יש לשחרר את הזיכרון שהוקצה ולא ניתן להשתמש בו (אם יש כזה) ולהחזיר NULL.

2.

Apartment* DuplicateApartment (const Apartment* source);
הפונקציה מקבלת כפרמטר מצביע לדירה ומחזירה דירה חדשה (שמוקצית דינמית) בעלת נתונים זהים לדירה שהתקבלה כפרמטר. לא לשכוח לבצע הקצאות לכל המשתנים שזקוקים לכך, על מנת למשל ששם המשפחה יועתק ולא יצביע לאותו מקום בזיכרון כמו המקורי. במידה ואחת ההקצאות בפונקציה לא מצליחה יש לשחרר את הזיכרון שהוקצה ולא ניתן להשתמש בו (אם יש כזה) ולהחזיר NULL.

3.

Floor* AddApartment(Floor* fl, const Apartment* apartment);
הפונקציה מקבלת מצביע לקומה ודירה ומחזירה מצביע לקומה מעודכנת. במידה והמצביע fl מצביע ל-NULL, יש ליצור קומה חדשה עם השם "Normal", מספר קומה 2, ועם דירה אחת שתאותחל על ידי העתקה מתוכנה של הדירה apartment. לא לשכוח לבצע הקצאות לכל המשתנים שזקוקים לכך, על מנת למשל ששם המשפחה יועתק ולא יצביע לאותו מקום בזיכרון כמו המקורי. במידה ובקומה כבר יש דירה עם מספר זהה, הפונקציה רק תעדכן את שם המשפחה, כמות החדרים ואת השטח של הדירה הקיימת בקומה לפי שם המשפחה, כמות החדרים והשטח שהועברו ב-apartment.
במידה ובקומה אין דירה עם מספר כזה, נגדיל את מערך המצביעים של הקומה, נוסיף דירה חדשה ונעתיק את הנתונים מ-apartment לדירה החדשה. במידה וההקצאה הצליחה, יש לעדכן את השדה num_of_apartments.
במידה ואחת ההקצאות בפונקציה לא מצליחה יש לשחרר את הזיכרון שהוקצה ולא ניתן להשתמש בו (אם יש כזה) ולהחזיר NULL.

4.

Floor* DuplicateFloor(const Floor* source);
הפונקציה מקבלת מצביע לקומה קיימת source ועליה ליצור קומה חדשה מאותחלת בנתונים של source. יש לבצע הקצאה לכל המשתנים ולא העתקת מצביעים, כלומר להקצות לדירות זיכרון חדש וגם לכל הנתונים הדורשים הקצאה. במידה ואחת ההקצאות בפונקציה לא מצליחה יש לשחרר את הזיכרון שהוקצה ולא ניתן להשתמש בו (אם יש כזה) ולהחזיר NULL.

הערה: ניתן להשתמש לצורך המימוש בפונקציה מסעיף 2.

5.

Building* DuplicateBuilding(const Building* source);
הפונקציה מקבלת מצביע לבניין קיים source ועליה ליצור בניין חדש מאותחל בנתונים של source. יש לבצע הקצאה לכל המשתנים ולא העתקת מצביעים, כלומר להקצות לקומות זיכרון חדש וגם לכל הנתונים הדורשים הקצאה. במידה ואחת ההקצאות בפונקציה לא מצליחה יש לשחרר את הזיכרון שהוקצה ולא ניתן להשתמש בו (אם יש כזה) ולהחזיר NULL.

הערה: ניתן להשתמש לצורך המימוש בפונקציה מסעיפים 2 ו 4.

6.

Building* AddFloor(Building* building, const Floor* fl);
הפונקציה מקבלת מצביע לבניין ומצביע לקומה. במידה והבניין מצביע ל-NULL יש ליצור בניין חדש, עם שם רחוב "Rotshild" ומספרו יהיה 1. כמו כן הוא יהיה בעל קומה אחת שתאוחל ע"י העתקה מתוכנה של הקומה fl. במידה והבניין קיים, רק במידה וקומה זו לא קיימת במערך המצביעים (לפי מספר הקומה) יש להוסיפה (להגדיל את מערך המצביעים באמצעות הקצאה דינאמית ולהוסיף העתק של הקומה fl), לעדכן את המצביע לבניין ולהחזיר את המצביע העדכני. במידה ואחת ההקצאות בפונקציה לא מצליחה יש לשחרר את הזיכרון שהוקצה ולא ניתן להשתמש בו (אם יש כזה) ולהחזיר NULL.

עם סוף 6
יציאת
אין

7.

void SortByName(Floor* fl);
הפונקציה מקבלת מצביע לקומה fl וממיינת את הדירות בקומה לפי שם המשפחה לקסיקוגרפית מהקטן לגדול.

8.

void SortByNumber(Floor* fl);
הפונקציה מקבלת מצביע לקומה fl וממיינת את הדירות בקומה לפי מספר הדירה מהקטן לגדול.

9.

float GetArea(const Floor* fl, const char* fname);
הפונקציה מקבלת מצביע לקומה fl ושם משפחה fname. הפונקציה מחפשת את הדירה עם השם משפחה הזה בקומה. במידה ומוצאת מחזירה את שטח הדירה, במידה והדירה לא נמצאת, הפונקציה תחזיר -1.

10.

float GetTotalRooms(Building* building);
הפונקציה מקבלת מצביע לבניין ומחזירה כמה חדרים יש בכל הדירות בכל הקומות.

11.

void FreeApartment(Apartment* apartment);
הפונקציה מקבלת מצביע לדירה ומשחררת את הזיכרון שהדירה משתמשת בו.

12.

void FreeFloor(Floor* fl);
הפונקציה מקבלת מצביע לקומה ומשחררת את כל נתוני הקומה בצורה עמוקה (ניתן להשתמש בפונקציה מסעיף 11).

13.

void FreeBuilding(Building* building);
הפונקציה מקבלת מצביע לבניין ומשחררת את כל נתוני הבניין בצורה עמוקה (ניתן להשתמש בפונקציות מסעיפים 11,12).

אסוף 7
מאפיין
והיונים
עשיתי
עם קורס
אסוף 8
בתחילת