



מבוא למדעי המחשב

מועד א – תשפ"א (סמסטר א)

קוד נושא: 611264

תאריך הבחינה: יום שלישי 23/2/2021

שעת תחילת הבחינה: 14:00

שעת סיום הבחינה: 18:00 (זמן זה כולל זמן התארגנות)

מרצים: ד"ר גלית חיים, ד"ר יהודה אלמליח

מתרגלים: מר אמיר ישורון, גב' תמר ריזמוביץ, גב' ענבל אברהם, מר אביב שוקרון

הקדמה

במבחן זה עליכם לענות על 4/4 שאלות תכנותיות ב C. משך המבחן 4 שעות. עליכם להקפיד היטב על ההוראות, ובפרט על הוראות ההגשה, שכן הבדיקה הינה אוטומטית (למעט השאלה על המבנים).

אתם מקבלים:

- ארבעה קובצי c וקובץ h אחד המכיל את כל החתימות הנדרשות.
- אין להוסיף include-ים כלשהם וגם אין בכך צורך.
- עליכם להשלים את הקוד בקובצי ה c בלבד, ע"פ הגדרות השאלה
- בנוסף, אתם מקבלים את קובץ ה mainTrain המכיל בדיקות לוגיות. זהו ה main של מוד האימון. הוא חשוף, וייתן לכם מושג לגבי הבדיקות הלוגיות של המבחן.
- שימו לב שהוא אינו בודק את *כל* הבדיקות הלוגיות של המבחן, לכן עליכם להתרכז בהוראות המבחן ולא רק לגרום ל mainTrain להצליח.

עליכם להגיש:

- את קבצי ה c מושלמים. לא ב zip או דומיו, אלא את קבצי המקור עצמם.

שימו לב: לא יתקבל כל ניקוד לשאלה שעבורה הקוד לא מתקמפל או שיש לו שגיאות בזמן ריצה

הבדלים בין מוד אימון להגשה:

מוד אימון	מוד הגשה
בדיקה חשופה המכילה חלק מהבדיקות	בדיקה חסויה המכילה את כל הבדיקות
ניתן להגיש אינספור פעמים עד לתום הבחינה	ניתן להגיש אינספור פעמים עד לתום הבחינה
משוב מלא – שגיאות קומפילציה, ריצה ולוגיות	משוב חלקי – שגיאות לוגיות בלבד
ציון מידי. זהו אינו ציון המבחן ולא חלק ממנו	ציון מידי – זהו ציון המבחן בגין שלושת השאלות הראשונות



לא נחשב כהגשת המבחן (הגשת המבחן תיעשה במוד "הגשה סופית")	לא נחשב כהגשת המבחן
--	---------------------

שימו לב! רק שאלות 1-3 נבדקות אוטומטית, ואילו שאלה 4 נבדקת ידנית. במידה ולא קבלתם את מלוא הנקודות על שאלה מספר 3 היא תבדק ידנית.

היות ושאלה 4 לא נבדקת בזמן הבחינה, במערכת הבדיקות הציון המקסימלי שניתן לקבל הוא 60. ציון זה משמעותו קבלה של כל הנקודות על שאלות 1-3. לאחר הבדיקה הידנית של שאלה 4 (ושאלה 3 למי שלא קיבל את מלוא הנקודות) יפורסם לכם ציון סופי למבחן.

עם תחילת המבחן ניתן יהיה להוריד את קבצי הבחינה ממערכת הבדיקות ויחולקו הטפסים. האינטרנט ינותק והמבחן יתחיל. אל תחכו לרגע האחרון להגיש את המבחן בלחץ, ואז לגלות ששכחתם משהו. תכננו את הזמן היטב.

הגשה סופית: לאחר הגשת הבחינה (במוד הגשה סופית) לא תוכלו להגיש יותר את הבחינה. מיד בתום ההגשה הסופית תקבלו מספר אסמכתא בעל 4 ספרות. העתיקו את המספר ומסרו אותו לבוחנת.

בנוסף, עליכם ללחוץ על הקישור "לינק לשרת הגיבוי" ולהגיש את הפתרון שלכם גם בשרת הגיבוי.

קישור להורדת הבחינה: [/https://cktest.cs.colman.ac.il](https://cktest.cs.colman.ac.il)

הוראות הגשה

את המבחן יש להגיש ל cs_intro_semester_A | מועד א. יש למלא ת.ז עם 9 ספרות ולהוריד את המטלה.

- בכל הגשה יש להגיש את כל הקבצים Q1.c, Q2.c, Q3.c, Q4.c ורק אותם. גם אם עדין לא כתבתם את התשובות לכל השאלות עדין יש להגיש את כל הקבצים. כל שאלה נבדקת בנפרד. שימו לב לשמות הקבצים ("Q" גדולה).
- שים לב שצריך להגיש גם את Q4.c למרות שאינה נבדקת אוטומטית.
- שים לב שהציון המקסימלי הוא 60 כי שאלה 4 השווה 40 נק' תיבדק ידנית.
- רק הגשה במוד "הגשה סופית" תיחשב כהגשת המבחן.
- אחרי הגשה במוד "הגשה סופית" יתקבל מס' אסמכתא בן 4 ספרות (לא ניתן יהיה להגיש את הבחינה שוב). את מס' האסמכתא יש לשמור ולתת לבוחנת.

- חומר עזר: אין חומר עזר

- סטודנט/ית עם התאמת תוספת זמן, התוספת מצויה במערכת ותקבלו אותה באופן אוטומטי.

במידה ויתעוררו שאלות בנוגע לבחינה אתם מוזמנים לשלוח את השאלות באימייל לצוות הקורס בזמן הבחינה

לכתובת: **intro2cscolman@gmail.com**

הנחיות בדבר יושרה אקדמית

פעולה המנוגדת ליושרה האקדמית היא מעשה תרמית; הונאה. היא "גניבת דעת" תרתי משמע.

כל פעולה כזו מהווה פגיעה בערכים ובעקרונות היסוד של האקדמיה ובהם טוהר הבחינות, ותחושת הצדק והשוויון בין הסטודנטים/ות.



כל פעולה המנוגדת ליושרה האקדמית מבזה את הסטודנט/ית, פוגעת בכבודו/ה העצמי כאדם ובכבוד עמיתיו/ה הסטודנטיות והסטודנטים. לפיכך, על כל התשובות להיות מקוריות, להיכתב במלואן על ידי הסטודנט/ית במילותיו/ה, מבלי להיעזר ומבלי להתייעץ באחר/ת בשום דרך שהיא, ומבלי לקבל הערות לתשובות. (אלא אם מדובר בעבודת-בית שההנחיות הספציפיות לה יתירו במפורש עבוד ת-צוות).

על כן, חל איסור לשתף בתשובות איש/ה, וכן להעיר, ליעץ או לסייע בכל דרך אחרת לאחר/ת להונות במבחן.

על התשובות להיות מבוססות על חומרי העזר המותרים בלבד וחל איסור לבססן, במלואן או בחלקן, על תשובות של תלמידים אחרים, או להעתיקן מחומרים כלשהם כגון מאמרים, ספרים או מקורות אחרים, דיגיטליים או מודפסים. חלה חובה לציין כל ציטוט ואזכור הלקוחים ממקור חיצוני כלשהו.

על הבחינה/העבודה להיות מוגשת בזמן המפורש לכך, מתחייב מעקרון השוויון בין הסטודנטים/ות.

יש לציין כי המכללה עושה שימוש באמצעים שונים לגילוי פעולות המנוגדות ליושרה האקדמית לרבות גילוי העתקות, וכי הפרת כללי היושרה האקדמית מהווה לא רק ביזוי עצמי אלא גם עבירת משמעת ותטופל בהתאם.

בהצלחה!



שאלה 1: 20 נקודות

ממשו את הפונקציה:

```
int maxSumSubArr(int arr[], int arrSize);
```

הפונקציה מקבלת מערך חד מימדי ואת גודלו, ומחזירה את הסכום המקסימלי מבין כל תתי הסדרות האפשריות במערך. תת הסדרה תוגדר כאוסף ערכים רציפים. (ללא דילוגים).

דוגמא:

עבור המערך הבא: {8, 3, 8, -5, 4, 3, -40, 3, 5} הסכום המקסימלי שיוחזר יהיה 21. (תת הסדרה הרציפה עם הסכום המקסימלי היא {8, 3, 8, -5, 4, 3}).

סעיף א

הצלחה במוד אימון מזכה ב-6 נקודות

סעיף ב

הצלחה במקרי קצה במוד הגשה מזכה ב 6 נקודות

סעיף ג

הצלחה במוד הגשה בשאר המקרים מזכה ב-8 נקודות

הערה:

הציון המלא המכיל את כל הסעיפים הללו ינתן לאחר הגשה במוד הגשה.

ממשו את הקוד הנדרש בקובץ **Q1.c**



שאלה 2: 20 נקודות

ממשו את הפונקציה:

```
char * cleanExcessBlanks(char *text);
```

הפונקציה:

- מקבלת משפט בפרמטר text.

- מוחקת רווחים עודפים בין מילים, כאשר:

מילה: אוסף רציף של תווים כלשהם למעט רווח.

רווחים עודפים משמע: יותר מרווח אחד בין מילה למילה.

- מחזירה את המשפט המעודכן ללא רווחים עודפים.

אין צורך להקצות שטח זיכרון חדש בשביל המשפט המוחזר. יש לעדכן את text בהתאם.

דוגמאות:

עבור הקלט

Hi, What's up?

נקבל את הפלט

Hi, What's up?

עבור הקלט

I wish u Good Luck!@

נקבל את הפלט (הקלט לא השתנה. בין מילה למילה יש רווח אחד בלבד [אין רווחים עודפים])

I wish u Good Luck!@

סעיף א

הצלחה במוד אימון מזכה ב-6 נקודות

סעיף ב

הצלחה במקרי קצה במוד הגשה מזכה ב 6 נקודות

סעיף ג

הצלחה במוד הגשה בשאר המקרים מזכה ב-8 נקודות

הערה:

הציון המלא המכיל את כל הסעיפים הללו ינתן לאחר הגשה במוד הגשה.

ממשו את הקוד הנדרש בקובץ **Q2.c**



שאלה 3: 20 נקודות

כתוב פונקציה רקורסיבית:

```
int isSuminArr(int arr[], int size, int sum);
```

המקבלת מערך של מספרים שלמים, את גודלו ומספר sum.

הפונקציה בודקת האם קיימת תת סדרה של המערך שסכומה sum.

אם כן, הפונקציה תחזיר 1, אחרת 0.

דוגמאות:

- עבור המערך {5, 3, -40, 3, 4, -5, 8, 3, 8} והסכום 3- יוחזר 1, שכן קיימת תת הסדרה כזו: סכום האיברים 8 ו-5 הוא 3-
- עבור המערך {5, 3, -40, 3, 4, -5, 8, 3, 8} והסכום 6 יוחזר 1, שכן קיימת תת סדרה כזו: סכום האיברים 3, -5, 8 הוא 6.
- עבור המערך {9, -5, 7, 2} והסכום 1 יוחזר 0, שכן לא קיימת תת סדרה העונה להגדרות

הערות:

- אין להשתמש בלולאות, פונקציות עזר ו\או מערכי עזר.
- אין להשתמש במשתנים גלובאליים ו\או סטטיים.
- לכל סדרה קיימת תת סדרה ריקה. סכום סדרה ריקה הוא אפס.

סעיף א

הצלחה במוד אימון מזכה ב-6 נקודות

סעיף ב

הצלחה במקרי קצה במוד הגשה מזכה ב 6 נקודות

סעיף ג

הצלחה במוד הגשה בשאר המקרים מזכה ב-8 נקודות

הערה:

- הציון המלא המכיל את כל הסעיפים הללו ינתן לאחר הגשה במוד הגשה
- במידה והתשובה לשאלה זו לא תקבל את מלוא הנקודות התשובה תבדק ידנית.

את הפתרון לשאלה זו יש לכתוב בקובץ Q3.c

שאלה 4 (40 נק')

חברת Lime הינה חברה המציעה אופניים וקורקינטים שיתופיים. החברה נוסדה בשנת 2017 וגייסה בשנה זו \$12M. הרעיון השיתופי צבר תאוצה ואפשר לחברה בשנת 2018 לגייס \$335M (לפי שווי של 1.1 מליארד דולר). כיום ניתן לשכור



קורקינטים של ליים במספר ערים בארץ (וכמובן בעולם). בעזרת אפליקציה אותה מתקין הלקוח/משתמש הוא יכול למצוא איזה קורקינט נמצא בקרבת מקום (בעזרת שימוש ב-GPS) ולשחרר את הקורקינט לשימוש האישי ע"י סריקת ברקוד המופיע על הקורקינט.

בשאלה זו עליך לממש מספר פונקציות שימושיות עבור חברת Lime (בקובץ Q4.c)

להלן המבנים הנדרשים לאפליקציה:

מיקום (Location)

○ קורדינאטת מיקום בדי מימד לפי x ו- y

קורקינט (Scooter)

- id: מס' זיהוי יחודי של הקורקינט
- location: מיקומו הנוכחי של הקורקינט
- batteryLevel: מצב הסוללה – מס' בין 0 ל-5 המעיד על עוצמת הסוללה (5 – סוללה מלאה)
- status: מצב הקורקינט כמוגדר להלן:
 - 1 מעיד כי הקורקינט פנוי
 - 2 מעיד כי הקורקינט אינו פנוי
 - 3 מעיד כי הקורקינט אינו תקין

נסיעה (Ride):

- date: תאריך הנסיעה. 2 תווים ליום, 2 תווים לחודש וארבעה תווים לשנה (בתוספת '0\')
- סה"כ 9 תווים
- from: מיקום מהיכן נלקח הקורקינט לנסיעה
- to: מיקום (יעד) להיכן נסע הקורקינט
- scooterId: מספר זיהוי של קורקינט אשר ביצע את הנסיעה הנ"ל
- minutes: משך הנסיעה בדקות
- clientId: מס' זהות של הלקוח שביצע את הנסיעה
- price: מחיר הנסיעה

לקוח (Client):

- fname: שם פרטי
- lname: שם משפחה
- id: ת.ז. של הלקוח
- location: מיקום הלקוח
- numRides: מס' נסיעות שביצע עד כה ב-Lime
- rides: מערך מצביעים דינאמי לנסיעות שביצע הלקוח



ליים (Lime): המבנה המרכזי בו כל המידע של החברה מרוכז

- numScooters : מס' הקורקינטים בבסיס הנתונים של החברה
- scooters : מערך מצביעים (דינאמי) לקורקינטים של החברה
- numRides : מס' כל הנסיעות שבוצעו בעזרת הקורקינטים של ליים
- rides : מערך מצביעים (דינאמי) לנסיעות שבוצעו בקורקינטים של ליים
- numClients : מספר לקוחות/משתמשי החברה
- clients : מערך מצביעים (דינאמי) לכלל לקוחות/משתמשי החברה

להלן הגדרת המבנים:

```
typedef struct
{
    int x, y;
}Location;

typedef struct
{
    int id;
    Location location;
    int batteryLevel; //range 0-5
    char status; //1- available, 2- not available, 3- broken
}Scooter;

typedef struct
{
    char date[9]; // ddmmyyyy e.g. 25032020 means 20/03/2020
    Location from, to;
    int scooterId;
    int minutes;
    int clientID;
    float price;
}Ride;

typedef struct
{
    char fname[20];
    char lname[20];
    int id;
    Location location;
    int numRides;
    Ride** rides;
}Client;

typedef struct
{
    int numScooters;
    Scooter** scooters;
    int numRides;
    Ride** rides;
    int numClients;
    Client** clients;
}Lime;
```




סעיף א (20 נקודות)

ממש את הפונקציה הבאה:

```
Scooter** findRelevantScooters(Lime* lime, Client* client,  
                                int maxDistance, int minimumBattery, int* numAvailable);
```

הפונקציה מקבלת (לפי הסדר הבא) מצביע לליים - lime, מצביע ללקוח - client, מרחק מקסימאלי - maxDistance, עוצמת סוללה מינמאלית - minimumBattery, מצביע למספר numAvailable אותו תמלא הפונקציה (יוסבר בהמשך).

על הפונקציה להחזיר מערך מצביעים (אותו תקצה) לקורקינטים רלוונטיים העומדים בקריטריונים, את גודל המערך הפונקציה תמלא בפרמטר שהתקבל numAvailable.

קורקינט עומד בקריטריונים אם כל התנאים הבאים מתקיימים:

- מרחק הלקוח (client) מהקורקינט קטן שווה ל-maxDistance
- מצב הסוללה של הקורקינט גדולה שווה ל-minimumBattery
- הקורקינט פנוי

הקלה: ניתן להניח כי כל ההקצאות מצליחות.

תזכורת – אל תשכחו את פיתגורס בחישוב המרחק בין הלקוח לקורקינט.

כדי לחשב חזקה/שורש תוכלו להשתמש בפונקציה pow אשר נמצאת ב-math.h (לא לשכוח לבצע include בקובץ ה-c ולא ב-h. להלן חתימת pow:

```
double pow(double x, double y)
```

פונקציה זו מחזירה את הערך של x בחזקת y.

שאלה זו תבדק ידנית!



סעיף ב (20 נקודות)

ממש את הפונקציה הבאה:

```
void addNewRide(Lime* lime, Client* client, int scooterID, Location
destination, int minutes, float pricePerMinute,
char* date);
```

הפונקציה מקבלת (לפי הסדר הבא) מצביע לליים - lime, מצביע ללקוח - client, מספר מזהה של קורקינט - scooterID, יעד נסיעה - destination, משך הנסיעה בדקות - minutes, מחיר לפי דקה - pricePerMinute (יוסבר מטה) ומחרוזת תאריך - date.

על הפונקציה להקצות נסיעה חדשה על סמך הפרמטרים שהתקבלו. את הנסיעה יש להוסיף למערך המצביעים של lime וכן להוסיפה ללקוח (client) אשר מבצע את הנסיעה.

הפרמטרים שהתקבלו מגדירים את הנסיעה:

- הלקוח client ביצע את הנסיעה
- הנסיעה בוצעה בקורקינט שמספרו scooterID
- יעד הנסיעה הינו destination (כמובן שנקודת התחלת הנסיעה הינו מקום הקורקינט שמבצע את הנסיעה)
- הנסיעה בוצעה במשך minutes דקות
- הנסיעה בוצעה בתאריך date.

שים לב, מחיר הנסיעה מחושב כך: 5 ש"ח בגין נסיעה חדשה ועוד מחיר לדקה (pricePerMinute) כפול משך הנסיעה בדקות (ז"א $pricePerMinute * minutes + 5$).

הקלה: ניתן להניח כי כל ההקצאות מצליחות

שאלה זו תבדק ידנית

בהצלחה