



מבוא למדעי המחשב מועד א – תש"פ

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

מתרגלים: מר מתן דוגות, 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 דומין, 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	לא נחשב כהגשת המבחן (הגשת המבחן תיעשה במוד "הגשה סופית")

שימו לב! רק שאלות 1-3 נבדקות אוטומטית, ואילו שאלה 4 נבדקת ידנית. לכן 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 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 **(במודד 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://ck.cs.colman.ac.il/test.jsp>

חומר עזר: אין חומר עזר

בהצלחה!



שאלה 1: 20 נקודות

ממשו את הפונקציה:

```
void crossSort(int mat[][N], int numRows)
```

הפונקציה מקבלת מערך דו מימדי mat בעל numRows שורות ו-N (מוגדר ב-define) עמודות. על הפונקציה למיין בסדר עולה את השורות הזוגיות (שורה אפס נחשבת זוגית) ובסדר יורד את השורות האי זוגיות.

עבור המערך הבא:

1	8	3	3	4
7	12	4	0	84
91	10	5	4	7
14	19	10	10	0

נקבל:

1	3	3	4	8
84	12	7	4	0
4	5	7	10	91
19	14	10	10	0

ניתן להוסיף פונקציות עזר אך אין לשנות את הקובץ h הניתן לכם

ממשו את הקוד הנדרש בקובץ Q1.c



שאלה 2: 20 נקודות

כתבו פונקציה אשר מקבלת משפט בפרמטר `inStr` ומחזירה את המשפט לאחר השינוי הבא: כל מילה במשפט תיכתב בכיוון הפוך. (יש צורך להקצות מקום בשביל המשפט המוחזר).

- מילה = רצף של אותיות (גדולות או קטנות) בלבד

חתימת הפונקציה תהיה:

```
char *revWord(char inStr[]);
```

דוגמאות:

עבור הקלט

This is an example!34abc

נקבל את הפלט

sihT si na elpmaxe!34cba

עבור הקלט

Hi, What's up?

נקבל את הפלט

iH, tahW's pu?

הבהרה: שימו לב כי רק מילים מתהפכות ואילו שאר הטקסט שאינו מילה (כפי שהוגדרה) נשאר במקומו.

ממשו את הקוד הנדרש בקובץ **Q2.c**



שאלה 3: 20 נקודות

כתוב פונקציה רקורסיבית:

```
int MultEven(int a[], int size)
```

המקבלת מערך של מספרים שלמים (הגדולים שווים לאפס) ואת גודלו, ומחזירה את מכפלת המספרים הזוגיים במערך.

למשל עבור המערך

9	2	4	6	1	3
---	---	---	---	---	---

הפונקציה תחזיר 48 (כי $2 \cdot 4 \cdot 6 = 48$).

הערה:

במידה ולא קיימים איברים זוגיים במערך יש להחזיר 0.

הערות:

אין להשתמש בלולאות, פונקציות עזר ולאז מערכי עזר

את הפתרון לשאלה זו יש לכתוב בקובץ Q3.c



שאלה 4(40 נק')

בשנת 2014 פתחו סטודנטים מהמכללה למנהל אפליקציה בשם PayBox. האפליקציה מאפשרת העברת כסף בין אנשים. המטרה המרכזית בשנת 2014 היתה לאפשר תשלומים באופן קבוצתי. כיום האפליקציה מאפשרת מגוון רחב של תשלומים בין אנשים שונים. בשאלה זו עליך לממש מספר פונקציות שימושיות עבור PayBox (בקובץ Q4.c). להלן שלושה מבנים מרכזיים:



טרנזקציה/העברה (Transaction): העברה כספית.

- from_ID : ת.ז. של לקוח המעביר כסף
- to_ID : ת.ז. של לקוח אליו מועבר הכסף
- amount: סכום הכסף המועבר
- date: תאריך ביצוע הטרנזקציה. התאריך הינו מערך בגודל 9, המייצג 2 תווים עבור יום, 2 עבור חודש, ארבעה תווים עבור שנה (וכמובן תו נוסף עבור '0')
- remark: הערה (דינאמית) אשר ניתן לתאר בה את ההעברה

לקוח (Client):

- full_name: שם מלא של הלקוח (דינאמי)
- id: ת.ז. של הלקוח
- financialBalance: יתרת כספית של הלקוח
- transactions: מערך מצביעים (דינאמי) לטרנזקציות אותן ביצע הלקוח או כאלו שהועברו ללקוח
- transSize: מס' האברים במערך המצביעים

פייבוקס (PayBox):

- clients: מערך מצביעים (דינאמי) לכלל לקוחות PayBox
- clientSize: מס' אברים במערך המצביעים
- transactions: מערך מצביעים (דינאמי) לכלל הטרנזקציות שבוצעו
- transSize: מס' אברים במערך המצביעים (של הטרנזקציות)

להלן הגדרת המבנים:

```
typedef struct
{
    int from_ID;
    int to_ID;
    float amount;
    char date[9]; //ddmmyyyy
    char* remark;
}Transaction;

typedef struct
{
    char* full_name;
    int id;
    float financialBalance;
```



```
Transaction** transactions;
int transSize;
}Client;

typedef struct
{
    Client** clients;
    int clientSize;
    Transaction** transactions;
    int transSize;
}PayBox;
```

סעיף א (20 נקודות)

ממש את הפונקציה `addTransaction`.

הפונקציה מקבלת מצביע למבנה `PayBox`, ת.ז. של לקוח המבצע העברה כספית (`fromID`), ת.ז. של לקוח המקבל את ההעברה (`toID`), סכום הכסף המועבר (`amount`), תאריך בו מבוצעת ההעברה (`date`), סכום מקסימום המותר ללקוח להעביר ביום (`amountLimit`) והערה אודות ההעברה (`remark`).

על הפונקציה לבדוק אם טרנזקציה היא חוקית ולאחר מכן להוסיפה למערכת.

חוקיות טנזקציה:

- הלקוחות הרלוונטיים קיימים במערכת `PayBox`
- לקוח לא יכול להעביר סכום כספי גדול מ `amountLimit` ביום מסויים. לקוח יכול לבצע מספר טרנזקציות ביום אך אינו יכול להעביר בסכום מצטבר גבוה מ-`amountLimit`

הוספת טרנזקציה:

- יש להוסיף טרנזקציה חדשה למערך הרלוונטי ב-`PayBox`
- יש לעדכן את הלקוחות הרלוונטיים במערך הטרנזקציות שלהם
- יש לעדכן את שדה `finacialBalance` של הלקוח בהתאם. במידה והלקוח קיבל סכום יש להוסיף את הסכום ל-`financialBalance` במידה והלקוח העביר סכום יש להוריד את הסכום מהשדה הנ"ל של הלקוח.

הפונקציה מחזירה 1 אם הטרנזקציה חוקית והיא התבצעה במלואה. במידה והטרנזקציה אינה חוקית יש להחזיר 0.

הקלה: ניתן להניח כי כל ההקצאות מצליחות

להלן חתימת הפונקציה:

```
int addTransaction(PayBox* pb, int fromID, int toID, float amount,
    char* date, float amountLimit, char* remark)
```



סעיף ב (20 נקודות)

מס הכנסה אינו נשאר אדיש נוכח ההעברות המתקיימות באופן תדיר ב-PayBox. בכדי למנוע העלמות מס, מבקש מס הכנסה מ-PayBox לממש עבורו את הפונקציה IrsReport.

```
Client** IrsReport(PayBox* pb, char* year, float amount, int* repSize);
```

הפונקציה הנ"ל מקבלת מצביע ל-PayBox (pb), מצביע לשנה (מערך תווים בעל 4 ספרות), סכום כסף (amount) ומצביע לגודל מערך שתחזיר (יוסבר בהמשך).

הפונקציה תחזיר מערך מצביעים (אותו תקצה) ללקוחות אשר כל אחד מהם בצע העברות בשנה year מעל סכום מצטבר amount.

גודל המערך שיוחזר יעודכן בפרמטר repSize אותו תזין הפונקציה.

ניתן להניח כי כל ההקצאות יצליחו.

בהצלחה



הוראות הגשה

בית הספר למדעי המחשב
מערכת הגשה ובדיקה אוטומטית תש"פ

הגשת בחינה

ת.ז. - 9 ספרות
סיסמה
קורס
שם המטלה
פניולה
קבצים

מועד א' ▼
cs_intro_semester_A ▼
הורדת המבחן
Choose Files No file chosen

שלח

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.